Passive Aero-Acoustic Sensor Self Interference Cancellation

Phase I Final Report

By Felix Rosenthal and J. Clarke Stevens,
Signal Separation Technologies

August 23, 1994

SIGNAL SEPARATION TECHNOLOGIES
4020 Iva Lane, Annandale, Virginia 22003
(703) 978-4973; Fax: Please voice first

Contract DAAA21-94-C-0025 for
U. S. Army ARDEC, Picatinny Arsenal,
SMCAR-FSF-RM
William Donnally, Project Officer

SIGNAL SEPARATION TECHNOLOGIES

4020 IVA LANE, ANNANDALE, VIRGINIA 22003 (703) 978-4973 Fax: Please voice first

September 26, 1994

Mr. Norman Walton
DTIC - OCC
Cameron Station
Alexandria, VA 22304-6145

Dear Mr. Walton:

Thank you for your call this morning. Enclosed is another copy of the Form 1473, to be inserted into our Final Report, Phase I under Army contract DAAA21-94-C-0025. I apologize for the original omission. Distribution of the report is unlimited.

If I can be of any further help, please let me know.

Very truly yours,

F. Rosenthal

Felix Rosenthal

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY: N.A.

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE: N.A.

3. DISTRIBUTION / AVAILABILITY OF REPORT:

4. PERFORMING ORGANIZATION REPORT NUMBER(S): Phase I Final Report

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: Signal Separation Technologies

6b. OFFICE SYMBOL (If applicable):

6c. ADDRESS (City, State, and ZIP Code): 4020 Iva Lane, Annandale, VA 22003

7a. NAME OF MONITORING ORGANIZATION:

7b. ADDRESS (City, State, and ZIP Code):

8a. NAME OF FUNDING / SPONSORING ORGANIZATION: U.S. Army ARDEC

8b. OFFICE SYMBOL (If applicable): SMCAR-FSF-RM

8c. ADDRESS (City, State, and ZIP Code): Picatinny Arsenal, NJ 07806-5000

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification): Passive Aero-Acoustic Sensor Self Interference Cancellation (U)

12. PERSONAL AUTHOR(S): Rosenthal, Felix and Stevens, J. Clarke

13a. TYPE OF REPORT: Final Report

13b. TIME COVERED: FROM 940223 TO 940823

14. DATE OF REPORT (Year, Month, Day): 940823

15. PAGE COUNT: 82

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): HMMWV vehicle, Armored personnel carrier, Detection, Classification, Air-acoustic, Noise cancel(-ling, -lation), Self-noise, Signal, Processing, Singular-value Decomposition, SFR-SVD Method

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Some Army HMMWV armored personnel carrier vehicles carry air-acoustic sensor systems comprised of a complement of roof-mounted microphones to detect and identify other vehicles and aircraft such as helicopters. The performance of these systems tends to become degraded as a result of the vehicle's own engine and road noise. Multi-channel noise cancellation, if correctly applied, is capable of removing most of the own-vehicle's engine self noise from the microphone outputs. The purpose of this project is to improve the performance of these acoustic systems through the application of multi-channel noise cancellation. The purpose of Phase I was to establish the efficacy of Signal Separation Technologies' SFR-SFC (Signal-free Reference using Singular-value Decomposition) technique for performing multi-channel noise cancellation for the aero-acoustic sensor application. This efficacy has been firmly established as a result of Phase I tests and analysis. In a typical example, a signal peak previously submerged in noise of the same level was made to stand out over the noise by some 15 dB. The SFR-SVD technique to do multi-channel noise cancellation is covered by U.S. Patent No. 5,209,237 and foreign patents. It is available to the U.S. Government for Government purposes only on a royalty-free basis. For all other purposes, a license must be negotiated with Signal Separation Technologies.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL: Felix Rosenthal

22b. TELEPHONE (Include Area Code): (703) 978-4973

22c. OFFICE SYMBOL:

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Contents

Abstract:

Some Army HMMWV armored personnel carrier vehicles carry air-acoustic sensor systems comprised of a complement of roof-mounted microphones to detect and identify other vehicles and aircraft such as helicopters. The performance of these systems tends to become degraded as a result of the vehicle's own engine and road noise. Multi-channel noise cancellation, if correctly applied, is capable of removing most of the own-vehicle's engine self noise from the microphone outputs. The purpose of this project is to improve the performance of these acoustic systems through the application of multi-channel noise cancellation.

The purpose of Phase I was to establish the efficacy of Signal Separation Technologies' SFR-SFC (Signal-free Reference using Singular-value Decomposition) technique for performing multi-channel noise cancellation for the aero-acoustic sensor application. This efficacy has been firmly established as a result of Phase I tests and analysis. In a typical example, a signal peak previously submerged in noise of the same level was made to stand out over the noise by some 15 dB.

Also as a result of having conducted the tests and analyzed the measurements, we have learned a great deal about vehicles and the effects of sensor placement.

The SFR-SVD technique to do multi-channel noise cancellation is covered by U.S. Patent No. 5,209,237 and foreign patents. It is available to the U.S. Government for Government purposes only on a royalty-free basis. For all other purposes, a license must be negotiated with Signal Separation Technologies.

Key Words:

HMMWV vehicle, Armored personnel carrier, Detection, Classification, Air-acoustic, Noise cancel(-ling, -lation), Self-noise, Signal Processing, Signal-free reference, Singular-value decomposition, SFR-SVD Method

I. Executive Summary

Some Army HMMWV vehicles carry air-acoustic sensor systems comprised of a complement of roof-mounted microphones to detect and identify other vehicles and aircraft such as helicopters. The performance of these systems tends to become degraded as a result of the vehicle's own engine and road noise. The purpose of this project is to improve the performance of these acoustic systems through the application of multi-channel noise cancellation.

The purpose of Phase I was to establish the efficacy of Signal Separation Technologies' SFR-SVD (Signal-free Reference using Singular-value Decomposition) patented and widely tested technique for accomplishing this task. SFR-SVD is the only known method for correctly performing multi-channel noise cancellation. Its efficacy for the present application has been firmly established as a result of Phase I tests and analysis. In a typical example, a signal peak previously submerged in noise of the same level was made to stand out over the noise by some 15 dB. Signal-to noise improvements of 20 dB and more have been common in the past when the SFR-SVD method has been applied. The purpose of a Phase II project to be proposed is to perform additional tests needed to identify improved sensor locations, particularly for the noise sensors, and then to develop a real time system to cancel the noise on line in real time.

Phase I began with an analysis of data previously taken by Lockheed Sanders, Inc. Since the available data did not contain signal, and since SST considered the measurement and analysis of signal an essential part of demonstrating the worth of our methods, this existing data was used primarily to prepare the software and data formats. New tests were conducted with Lockheed Sanders, Inc. at their facilities on June 2, 1994. The analysis of these test results is reported in this document.

A Phase II project will be proposed to identify and test improved locations of sensors, especially reference sensors. Assuming this effort to be successful, the software will be adapted and built into a real-time multi-channel noise cancelling system. Such a system would find applications, not only in the aero-acoustic sensor interference problem, but also in several other Army noise cancellation applications such as an endfire acoustic detection array under development at Army Research Laboratories, and an acoustic sensor project under development at the Night Vision Electronic Sensors Directorate. In addition, such new equipment technology will be highly transferable to commercial applications including the communications industry and to electroencephalography and other medical instrumentation.

The SFR-SVD technique to do multi-channel noise cancellation is covered by U.S. Patent No. 5,209,237 and foreign patents. It is available to the U.S. Government for Government purposes only on a royalty-free basis. For all other purposes, a license must be negotiated with Signal Separation Technologies.

II. Phase I Objectives and their Accomplishment

The purpose of this project is to improve the performance of Army vehicle-mounted air acoustic systems through the application of multi-channel noise cancellation to suppress self noise of the vehicle. The purpose of Phase I was to establish the feasilility of doing this using SST's patented SFR-SVD Method. This feasibility has been thoroughly established as a result of these tests, and the efficacy of the SFR-SVD algorithm and method has once again been reaffirmed, in the following senses:

(1) In every instance that has been examined, any **noise** which appeared in a signal sensor and also appeared substantially and coherently in one or more reference sensors, was recognized and **cancelled** in the signal sensors' corrected output.

(2) In every instance that has been examined, any **signal** which appeared in a signal sensor but did not appear substantially and coherently in any reference sensor applied to the cancellation, was **retained** in the corrected output.

(3) Conversely, in every instance in which signal was unfortunately found to be cancelled in the corrected output, it is demonstrated that this signal not only appeared substantially in the references but did so coherently with the signal stemming from the source. The resulting signal cancellation was exactly what was to be expected under the circumstances: when a reference sees substantial signal coherently with a signal sensor, then that signal will be interpreted as noise and get cancelled accordingly. As stated in SST's proposal leading to this project, signal in reference sensors to an extent greater than a quantifiable amount, say 5 or 10 percent of the largest noise source present there, cannot be tolerated. On the other hand, when signal entry into reference sensors is limited to that extent, then

it is readily kept out of the required noise cancellation filters by an appropriate setting of the SFR-SVD eigenvalue cutoff thresholds, and the signal will then be preserved intact as it should.

In most cases during the June 2 test series, substantial loudspeaker signal energy was later found to have entered all the references, which in principle are supposed to get primarily self-noise energy. Since this was the first such test series in which SST had participated, and since the primary purpose of the test was to verify the efficacy of our algorithms and software for this application, we did not feel justified in suggesting substantial changes of sensors or sensor locations from those which had been used in previous tests by Lockheed-Sanders, Inc. Moreover, during the June 2 tests our most prevalent concern was to get the loudspeakers close enough so that the microphones would hear them. The microphones on the far side of the vehicle from the loudspeakers did not hear the microphones as well as those on the near side, and would probably be better mounted standing up from the roof. But the main difficulty turned out to be too much signal in the references.

Apart from sensor locations, another factor which may have affected the various sensor coherences and hence caused signal cancellation was the fact that over most of the frequency range of interest, the HMMWV and its sensors stood in the near field of the loudspeakers. In that region, the acoustic waves are nonplanar and may contain other important nonlinearities. Since coherence has everything to do with whether a set of signals or noises are linearly related, this is not a trivial matter and must be more carefully controlled in subsequent tests.

In any case, the most important problem in the tests was the entry of too much signal into the references, compared to the noise levels they received. Perhaps it would have been better to run the HMMWV engine at a higher speed rather than idling, to increase the noise levels. We are now convinced that optimal types and locations of all sensors, but especially the references, need to be determined through a carefully designed experiment.

Thereore, the most important thing which remains to be done is to carry out such a carefully instrumented and controlled second experiment, in which source and noise levels entering all sensors can be subjected to thorough spectral and coherence analysis on-site, and sensor types and locations can be adjusted in order to optimize the "noise to signal ratio" in the references. It is demonstrated in Section VD of this report that monitoring the various power spectral densities may not be sufficient in this regard; the coherences of the sensors with the source must be known as well in order to assure intelligent choices for sensor locations. We are confident that efficacious reference sensors types and locations will be found as the result of such a test. We hope that such a new test would again be performed with the assistance of Lockheed Sanders, Inc., and wish to take full advantage of the valuable experience these people have already accumulated on this and related projects.

III. Theoretical Background

As stated in the proposal leading to this project, multi-channel noise cancellation in the context of separating signal from noise can only be accomplished correctly by using singular-value decomposition on the noise correlation or cross-spectral density matrix. This feature, which is the core of SST's multi-channel noise cancellation patent, is essential in order to eliminate all redundancies in the noise matrix, and in order to prevent signal cancellation when even the minutest amount of signals enter a noise sensor. But even when SFR-SVD is applied, signal must not enter the reference sensors at levels as large as those of the noise found there. When that happens, a threshold which is able to ignore the signal in the noise sensors, while utilizing the noise found there for filter calculations, is not attainable.

As has been stated in the proposal on which this project is based, effective multi-channel noise cancellation requires that the noise measurements used for cancellation must have two essential characteristics: (1) They must contain the noise to be cancelled coherently with the noise appearing in the signal sensors, and (2) they must contain no more than a modest amount of signal, relative to the noise that they receive. (There is some leeway in tolerable levels, but a signal level equal to the largest noise level would be excesssive, because then there would be no threshold which will keep noise but reject signal from the filter calculations). Analysis of coherences and power spectral densities calculated from the June 2 tests show that the first requirement was generally met, while the second one was not met to a sufficient degree. The result was that for the cases of greatest interest in the June 2 tests, the signal was cancelled along with the noise.

The code used for the analysis was a single-pass code, which calculates a single set of cross-spectral density matrices (CSD's), averaged over the entire time interval of interest. These matrices are used for the calculation of the Wiener filters that condition the noise before it is subtracted from the signal. The code is a modification of a single-pass code which has been used in the past for the analysis of batch mode fetal electrocardiograms. An alternative code would be an adaptive one, which continually updates the CSD's and also permits the continuous plotting of the corrected time series. The single-pass code was chosen for this analysis because of its greater simplicity and because it conveniently presents spectral information of signal and noise over a selected time span.

A word is in order on the meaning of "coherence." Coherence as plotted in the figures is the absolute cross spectral density between two sensors, normalized by the square root of the product of the two individual power (auto) spectral densities. Often in the literature, the coherence is taken as the square of this quantity. The reason we prefer to work with the square root of the usual function is that that square root is the complex Hermitian analogue of the readily visualized absolute-value cosine-of-the-included-angle between two vectors in an ordinary real n-dimensional vector space. It thus represents the degree of linear dependency vs. orthogonality between a pair of signals (or noises). Either way, as an absolute-value cosine or a cosine-squared, of course, the coherence is always between 0 and 1.

IV. Description of Tests

A day-long series of tests was conducted at Lockheed Sanders, Inc. facilities on June 2, 1994. Figure IV-1 is a photograph of the test vehicle standing in front of Lockheed Sanders, Inc.'s instrumentation shack. Figure IV-2 shows the loudspeaker and other instruments mounted on the back of a pickup truck, while Figure IV-3 depicts the front of the test vehicle showing the hood-mounted reference microphone. Figures IV-4 and IV-5 show the mounting of the tailpipe and front wheel well reference microphones respectively. The experimental setup is illustrated in Figure IV-6, showing the relative location of the HMMWV vehicle, its sensors, and the source loudspeaker system. Most of the instrumentation was located in a nearby instrument shed. The channel assignments listed in Figure IV-6 indicate the channels assigned to each of the 16 sensors: the 8 primary

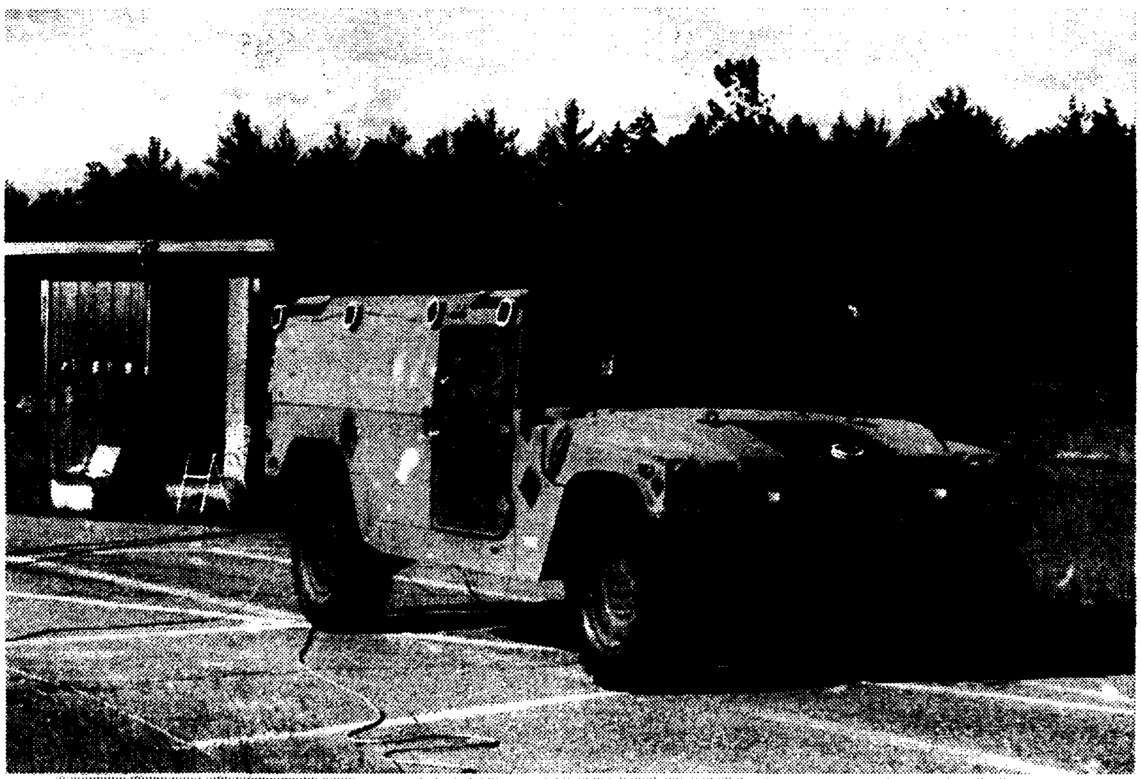


Figure IV-1. HMMWV test vehicle showing roof microphones, standing in front of Lockheed Sanders, Inc.'s instrumentation shack

Figure IV-2. Source loudspeaker and other equipment on back of pickup truck

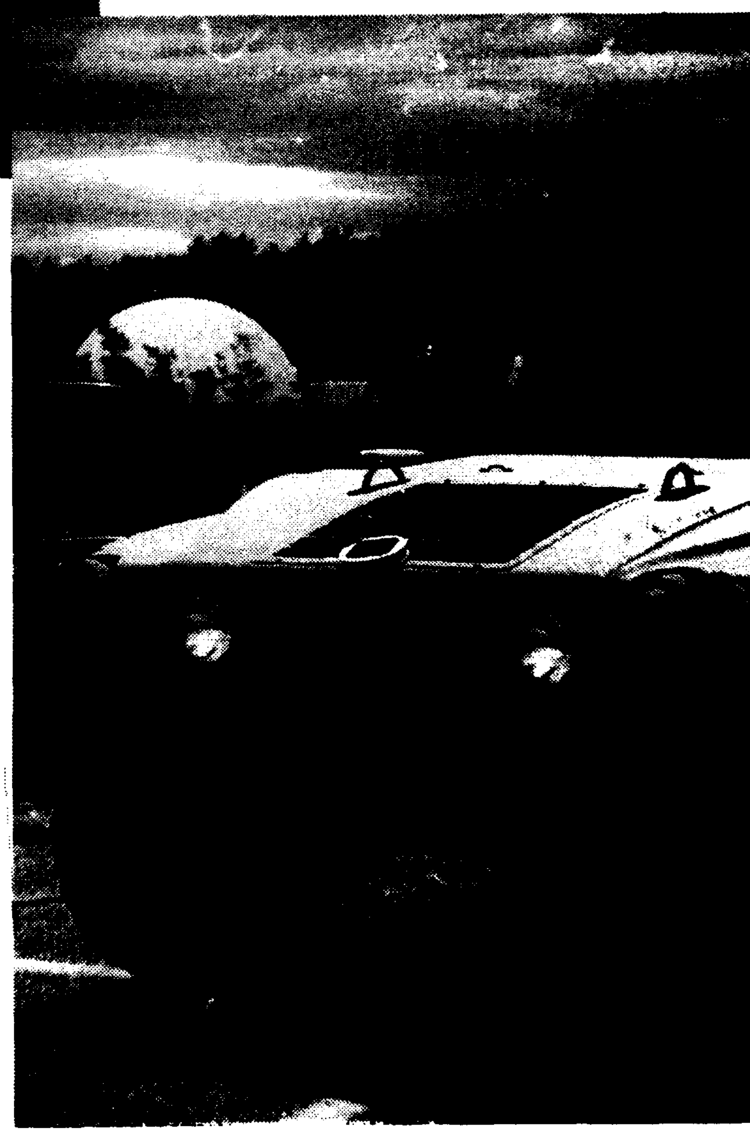


Figure IV-3. Front of test vehicle showing hood mounted microphone

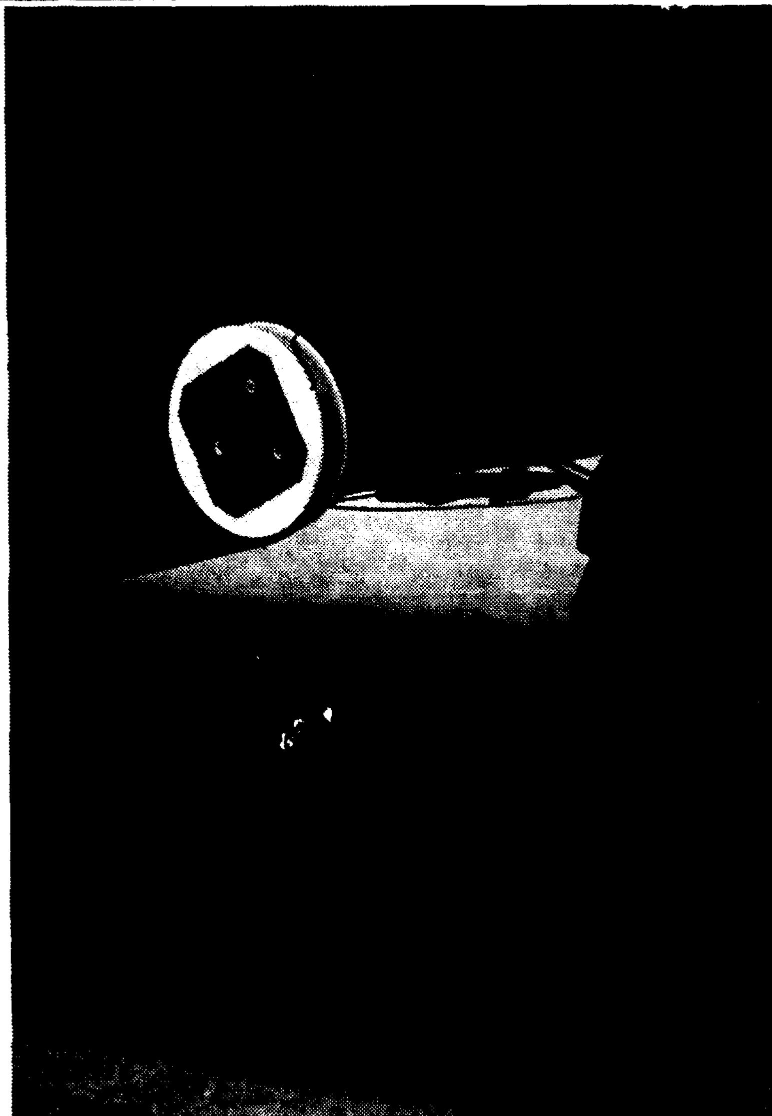


Figure IV-4. Tailpipe microphone

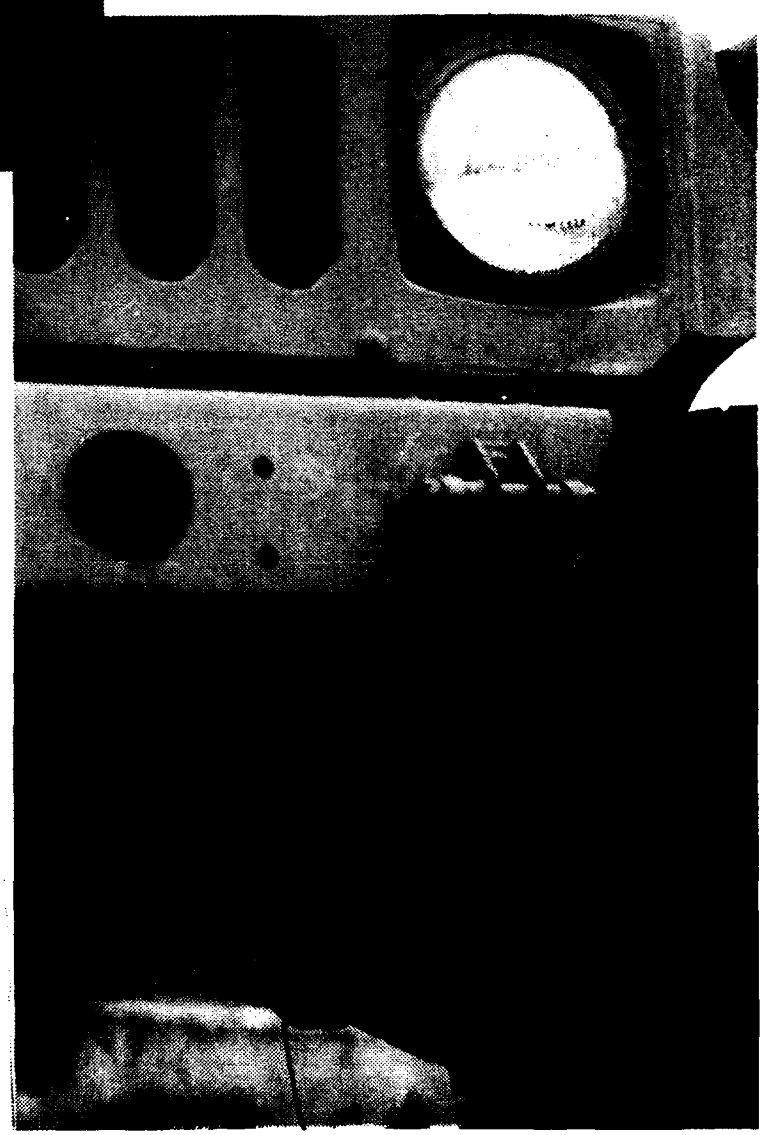
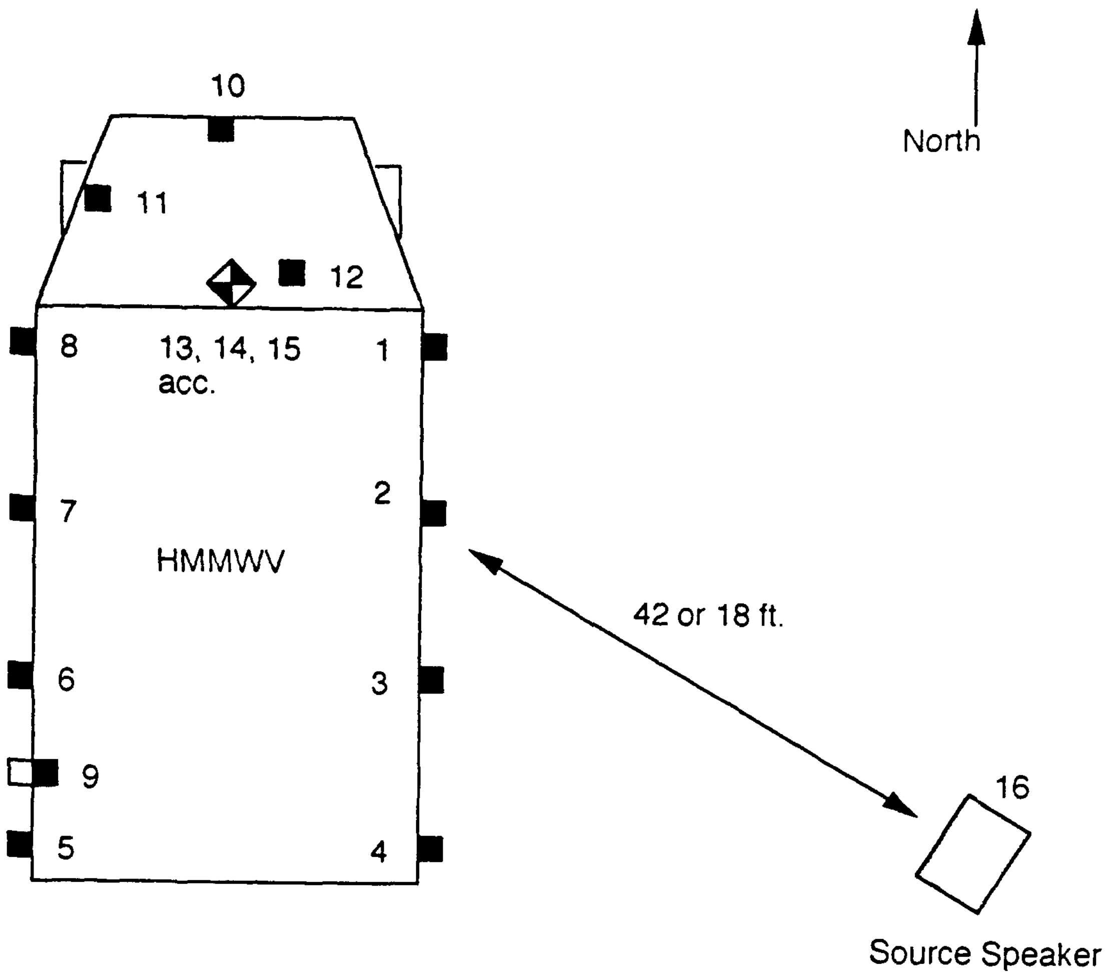


Figure IV-5. Wheel well microphone



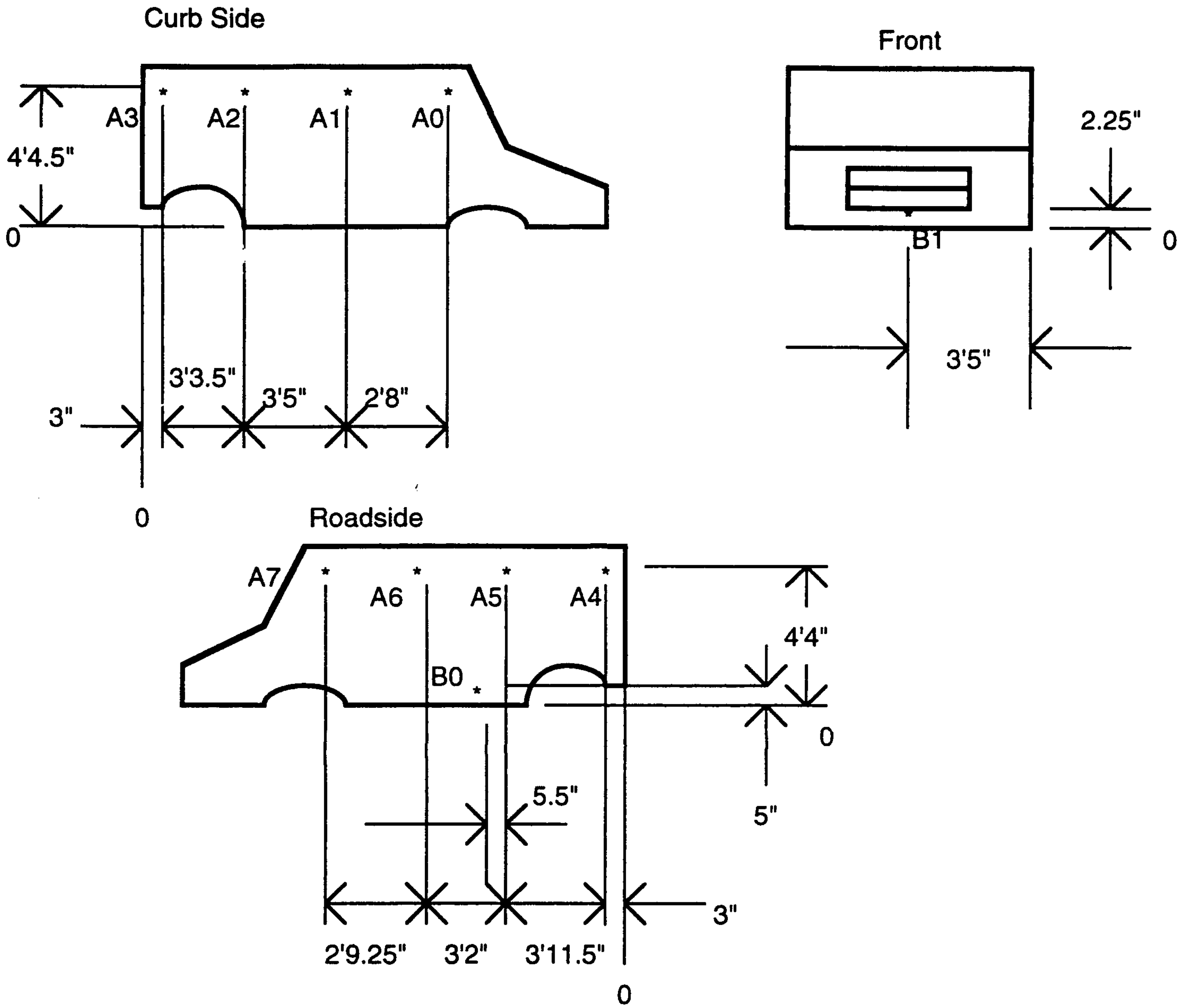
Channel Assignments

1. Right Side
2. Right Side
3. Right Side
4. Right Side
5. Left Side
6. Left Side
7. Left Side
8. Left Side
9. Tailpipe
10. Hood
11. Front Wheel Well
12. Engine
13. Acc. X axis
14. Acc. Y axis
15. Acc. Z axis
16. Reference Signal

Experimental Setup for measurements made at Lockheed-Sanders June, 2, 1994

Signal Separation Technologies
Felix Rosenthal and Clarke Stevens

Figure IV-6. Experimental setup for measurements at Lockheed-Sanders, Inc. June 2, 1994



Exterior mounting locations of the sensors on the HMMV

2 June 1994

Figure IV-7. Exterior mounting locations on the HMMWV, June 2, 1994

Calibration Chart for Accelerometer Type 4321

Brüel & Kjær

Serial No. 1384090

Reference Sensitivity at 50 Hz, 100 ms^{-2} and 23 C°

	Z	X	Y	Axis
Charge Sensitivity*	0,999	1,009	0,993	pC/ms^{-2}
Voltage Sensitivity*	0,774	0,851	0,850	mV/ms^{-2}
Capacitance (including cable)	1291	1186	1168	pF
Max. Transv. Sens.	1,2	2,0	1,6	%

Typical Capacitance of cable AO 0038 .... 110 pF

Typical Transverse Resonance Frequency ...... 14 kHz

Typical undamped natural frequency .... 50 kHz

Typical mounted resonance frequency .... 40 kHz

Polarity is positive on the center of the connector for an acceleration directed from the mounting surface into the body of the accelerometer

Resistance minimum 20000 MΩ at room temperature

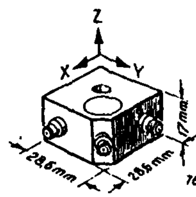
Date 87.10.22 Signature J. D.

1 g = 9,807 ms^{-2}

• This calibration is traceable to the National Bureau of Standards Washington D.C.

Environmental:
Humidity: Sealed
Temperature Range: –74 to +250°C (–100 to +482°F)
Max. Continuous Sinusoidal Acc. (peak): 5000 ms^{-2} or 500 g
Max. Shock Acceleration: 10000 ms^{-2} or 1000 g
Typical Magnetic Sensitivity (50 Hz): 4 ms^{-2}/T or 0,04 g/kgauss
Typical Temperature Transient Sensitivity: (Low Lim. Freq.: 3 Hz) 0,4 ms^{-2}/°C or 0,04 g/°C
Typical Base Strain Sensitivity: 0,02 ms^{-2}/μstrain or 0,002 g/μstrain
Specifications obtained in accordance with ANSI S2.11-1969

Physical:



Electrical Connectors: Coaxial 10 – 32 UNF-2A

Material: Titanium, ASTM Grade 2
Sensing Element: Piezoelectric, type PZ23
Weight: 55 gram
Construction: Delta Shear
Mounting Thread: 10 – 32 UNF-2B
Mounting Stud: 10 – 32 UNF-2A × 13 mm, steel stud or M4 × 16 mm steel bolt
Mounting Surface Flatness: < 3 μm
Mounting Torque: Normal 1,8 Nm. Min. 0,5 Nm. Max. 3,5 Nm
Seismic Mass: 2,6 gram
For further information see B & K "Piezoelectric Accelerometer and Preamplifier" handbook

Pat DK [illegible]1401

Figure IV-8. Calibration chart for Accelerometer type 4321

Sensor No.	Group	Sensor s/n	Description	Cal. Data	
0	A	6	Curb side front of door	1	
1	A	20020	Curb side rear of door	0.9	
2	A	20007	Curb side left of center	1.5	
3	A	20019	Curb side rear	0.7	
4	A	20024	Road side rear	1.4	
5	A	5	Road side right of center	1.1	
6	A	4	Road side rear of door	1.1	
7	A	20011	Road side front of door	1	
0	B	20003	ANC	1.4	
1	B	3	Hood	1.05	
2	B	20004	Road side front frame	1.25	
3	B	20008	Curb side frame under door	1.1	
4	B		Accelerometer "X"		
5	B		Accelerometer "Y"		
6	B		Accelerometer "Z"		
7	B	20005	Signal from Speaker	0.9	

Figure IV-9. Calibration data for Sensors

Run 1 (Tape 5)			
Engine off, reference directly connected			
Event	Offset	Real Time	Comment
Start tape	0:00:00	9:14:23	
1 kHz sine wave starts	0:01:00	9:15:23	
1 kHz sine wave stops	0:02:00	9:16:23	
500 Hz sine wave starts	0:03:00	9:17:23	
500 Hz sine wave stops	0:04:00	9:18:23	
250 Hz sine wave starts	0:05:00	9:19:23	
250 Hz sine wave stops	0:06:00	9:20:23	
25 Hz sine wave starts	0:07:00	9:21:23	
Gun shot	0:07:17	9:21:40	
jet low amplitude	0:07:37	9:22:00	
25 Hz sine wave stops	0:08:00	9:22:23	
100 Hz-1 kHz sine wave sweep starts	0:08:56	9:23:19	
100 Hz-1 kHz sine wave sweep stops	0:09:56	9:24:19	
sound file e04a2063 starts	0:10:56	9:25:19	
horns	0:12:37	9:27:00	
sound file e04a2063 stops	0:12:52	9:27:15	
sound file e04a2066 starts	0:14:52	9:29:15	
sound file e04a2066 stops	0:16:06	9:30:29	
sound file e04a2070 starts	0:18:06	9:32:29	
jet fly-over	0:19:37	9:34:00	
sound file e04a2070 stops	0:20:07	9:34:30	
prop plane fly-over	0:20:37	9:35:00	
sound file e04a2071 starts	0:21:57	9:36:20	
sound file e04a2071 stops	0:23:05	9:37:28	
Stop tape	0:23:09	9:37:32	
Run 2 (Tape 4)			
engine off, microphone on source speaker			
Event	Offset	Real Time	Comment
Start tape	0:00:00	10:01:00	
gun shots	0:01:00	10:02:00	
1 kHz sine wave starts	0:01:00	10:02:00	
1 kHz sine wave stops	0:02:00	10:03:00	
500 Hz sine wave starts	0:03:00	10:04:00	
500 Hz sine wave stops	0:04:00	10:05:00	
250 Hz sine wave starts	0:05:00	10:06:00	
250 Hz sine wave stops	0:06:00	10:07:00	
25 Hz sine wave starts	0:07:00	10:08:00	
25 Hz sine wave stops	0:08:00	10:09:00	
100 Hz-1 kHz sine wave sweep starts	0:08:56	10:09:56	
100 Hz-1 kHz sine wave sweep stops	0:09:56	10:10:56	
sound file e04a2063 starts	0:10:56	10:11:56	
sound file e04a2063 stops	0:12:52	10:13:52	

Table 1. Lockheed tape log, p. 1

sound file e04a2066 starts	0:14:52	10:15:52	
sound file e04a2066 stops	0:16:06	10:17:06	
sound file e04a2070 starts	0:18:06	10:19:06	
sound file e04a2070 stops	0:20:07	10:21:07	
sound file e04a2071 starts	0:21:57	10:22:57	
sound file e04a2071 stops	0:23:05	10:24:05	
Stop tape	0:23:09	10:24:09	
Run 3 (Tape 3)			
engine running, microphone on source speaker			
Event	**Offset**	**Real Time**	**Comment**
Start tape	0:00:00	10:26:00	
helicopter fly-over	0:01:00	10:27:00	
1 kHz sine wave starts	0:01:00	10:27:00	
1 kHz sine wave stops	0:02:00	10:28:00	
prop plane	0:02:00	10:28:00	
500 Hz sine wave starts	0:03:00	10:29:00	
500 Hz sine wave stops	0:04:00	10:30:00	
250 Hz sine wave starts	0:05:00	10:31:00	
250 Hz sine wave stops	0:06:00	10:32:00	
25 Hz sine wave starts	0:07:00	10:33:00	
25 Hz sine wave stops	0:08:00	10:34:00	
100 Hz-1 kHz sine wave sweep starts	0:08:56	10:34:56	
100 Hz-1 kHz sine wave sweep stops	0:09:56	10:35:56	
sound file e04a2063 starts	0:10:56	10:36:56	
sound file e04a2063 stops	0:12:52	10:38:52	
sound file e04a2066 starts	0:14:52	10:40:52	
sound file e04a2066 stops	0:16:06	10:42:06	
sound file e04a2070 starts	0:18:06	10:44:06	
sound file e04a2070 stops	0:20:07	10:46:07	
vehicle drive-by	0:21:30	10:47:30	
sound file e04a2071 starts	0:21:57	10:47:57	
sound file e04a2071 stops	0:23:05	10:49:05	
Stop tape	0:23:09	10:49:09	
Run 4 (Tape 2)			
engine off, source speakers moved to 18 ft. from vehicle, 2 amplifiers, HF and LF (20-100 Hz)			
Event	**Offset**	**Real Time**	**Comment**
Start tape	0:00:00	11:38:23	
1 kHz sine wave starts	0:01:00	11:39:23	
1 kHz sine wave stops	0:02:00	11:40:23	
500 Hz sine wave starts	0:03:00	11:41:23	
500 Hz sine wave stops	0:04:00	11:42:23	
Truck drive-by	0:04:37	11:43:00	
250 Hz sine wave starts	0:05:00	11:43:23	

Table 2. Lockheed tape log, p. 2

250 Hz sine wave stops	0:06:00	11:44:23	
25 Hz sine wave starts	0:07:00	11:45:23	
25 Hz sine wave stops	0:08:00	11:46:23	
100 Hz-1 kHz sine wave sweep starts	0:08:56	11:47:19	
100 Hz-1 kHz sine wave sweep stops	0:09:56	11:48:19	
sound file e04a2063 starts	0:10:56	11:49:19	
Jet fly-over	0:12:37	11:51:00	
sound file e04a2063 stops	0:12:52	11:51:15	
sound file e04a2066 starts	0:14:52	11:53:15	
sound file e04a2066 stops	0:16:06	11:54:29	
sound file e04a2070 starts	0:18:06	11:56:29	
sound file e04a2070 stops	0:20:07	11:58:30	
sound file e04a2071 starts	0:21:57	12:00:20	
sound file e04a2071 stops	0:23:05	12:01:28	
Stop tape	0:23:09	12:01:32	
Run 5 (Tape 1)			
engine running, source speakers moved to 18 ft. from vehicle, 2 amplifiers, HF and LF (20-100 Hz)			
Event	**Offset**	**Real Time**	**Comment**
Start tape	0:00:00	12:16:40	phantom power supply used
1 kHz sine wave starts	0:01:00	12:17:40	
prop plane	0:01:20	12:18:00	
1 kHz sine wave stops	0:02:00	12:18:40	
500 Hz sine wave starts	0:03:00	12:19:40	
500 Hz sine wave stops	0:04:00	12:20:40	
prop plane	0:04:20	12:21:00	
250 Hz sine wave starts	0:05:00	12:21:40	
250 Hz sine wave stops	0:06:00	12:22:40	
25 Hz sine wave starts	0:07:00	12:23:40	
25 Hz sine wave stops	0:08:00	12:24:40	
prop plane	0:08:20	12:25:00	
100 Hz-1 kHz sine wave sweep starts	0:08:56	12:25:36	
100 Hz-1 kHz sine wave sweep stops	0:09:56	12:26:36	
sound file e04a2063 starts	0:10:56	12:27:36	
sound file e04a2063 stops	0:12:52	12:29:32	
sound file e04a2066 starts	0:14:52	12:31:32	
sound file e04a2066 stops	0:16:06	12:32:46	Some source clipping
sound file e04a2070 starts	0:18:06	12:34:46	Some source clipping
sound file e04a2070 stops	0:20:07	12:36:47	Some source clipping
sound file e04a2071 starts	0:21:57	12:38:37	
sound file e04a2071 stops	0:23:05	12:39:45	
Stop tape	0:23:09	12:39:49	
Run 6 (???)			
engine off, source speakers moved to 18 ft. from vehicle, 2 amplifiers, HF and LF (20-100 Hz)			

Table 3. Lockheed tape log, p. 3

Event	Offset	Real Time	Comment
Start tape	0:00:00	12:42:00	phantom power supply used
1 kHz sine wave starts	0:01:00	12:43:00	
1 kHz sine wave stops	0:02:00	12:44:00	
500 Hz sine wave starts	0:03:00	12:45:00	
prop plane	0:03:00	12:45:00	
500 Hz sine wave stops	0:04:00	12:46:00	
250 Hz sine wave starts	0:05:00	12:47:00	
250 Hz sine wave stops	0:06:00	12:48:00	
25 Hz sine wave starts	0:07:00	12:49:00	
25 Hz sine wave stops	0:08:00	12:50:00	
100 Hz-1 kHz sine wave sweep starts	0:08:56	12:50:56	
100 Hz-1 kHz sine wave sweep stops	0:09:56	12:51:56	
sound file e04a2063 starts	0:10:56	12:52:56	
sound file e04a2063 stops	0:12:52	12:54:52	
sound file e04a2066 starts	0:14:52	12:56:52	
sound file e04a2066 stops	0:16:06	12:58:06	
sound file e04a2070 starts	0:18:06	13:00:06	
prop plane	0:19:00	13:01:00	
sound file e04a2070 stops	0:20:07	13:02:07	
sound file e04a2071 starts	0:21:57	13:03:57	
sound file e04a2071 stops	0:23:05	13:05:05	
Stop tape	0:23:09	13:05:09	

Table 4. Lockheed tape log, p. 4

sensors were recorded on the *.g1 file (ch. 1-8), while the seven references (9-15) and the source (16) were simultaneously recorded on the *.g2 file.

Figure IV-7 shows the mounting locations of the sensors used on the HMMWV on June 2, while Figure IV-8 describes the accelerometers used in the experiment. Figure IV-9 provides calibration data for all sensors. Group A sensors 0 to 7 correspond to the primary sensors 1-8 as used throughout this report; Group B sensors 0-6 denote the 7 reference channels 9-15, while sensor 7 provided the source signal, channel 16.

We made 5 test runs. The corresponding tapes were numbered in inverse order, starting with Tape 5. (s. Lockheed tape log, Tables 1 - 4). For tapes 5, 4, and 2 the HMMWV engine was turned off; only tapes 3 and 1 contain own-engine noise. Of the two tests containing own-engine noise, Tape 3 was improperly recorded; Tape 5 was therefore used for most of the analysis herein. Unfortunately, this tape was recorded with the loudspeakers at a shorter distance from the vehicle than was the improperly recorded tape; it is possible that Tape 3 would have contained somewhat smaller near-field effects.

For each tape, several sine wave notes, and recordings of several vehicle passbys were played on the loudspeakers according to the log schedule. There were a few "sounds of opportunity" such as gunshots and an airplane flying overhead.

V. Noise Cancellation Analysis

We processed both simulated and real data. The simulations verified software operation, and the analysis of the real data (pure tones or simulated drivebys broadcast over a loudspeaker system) showed that noise cancellation as well as signal preservation took place in exemplary fashion whenever the data supported it. Analysis of the data also served to prove that much of the time, far too much signal reached the reference sensors, resulting in substantial signal cancellation.

A good example of noise cancellation is given in Figure V-1, representing a 5.1 second run taken from Tape 1 or a where the loudspeakers were playing a 31 Hz tone. This tone is found 7 minutes and 35 seconds, or 0455 seconds, into Tape 1 or a, as identified by the label starting with the file number a0455 in the Figure's title. (Conventions used in labelling file names are described in Appendix A.)

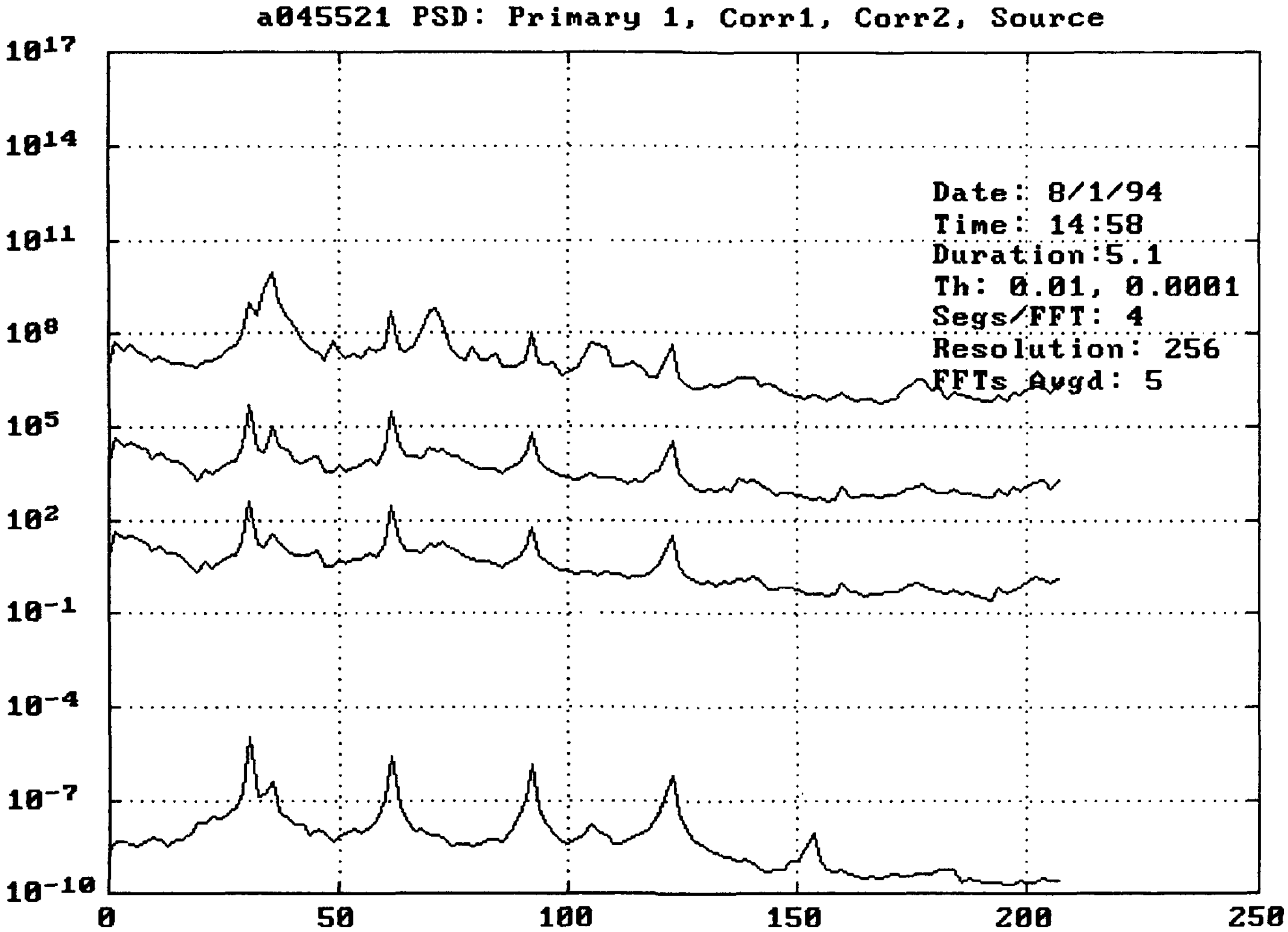


Figure V-1. Noise cancellation results for 25 Hz tone using 4 of 7 references: Uncorrected, Corrected a, Corrected b, and Source.

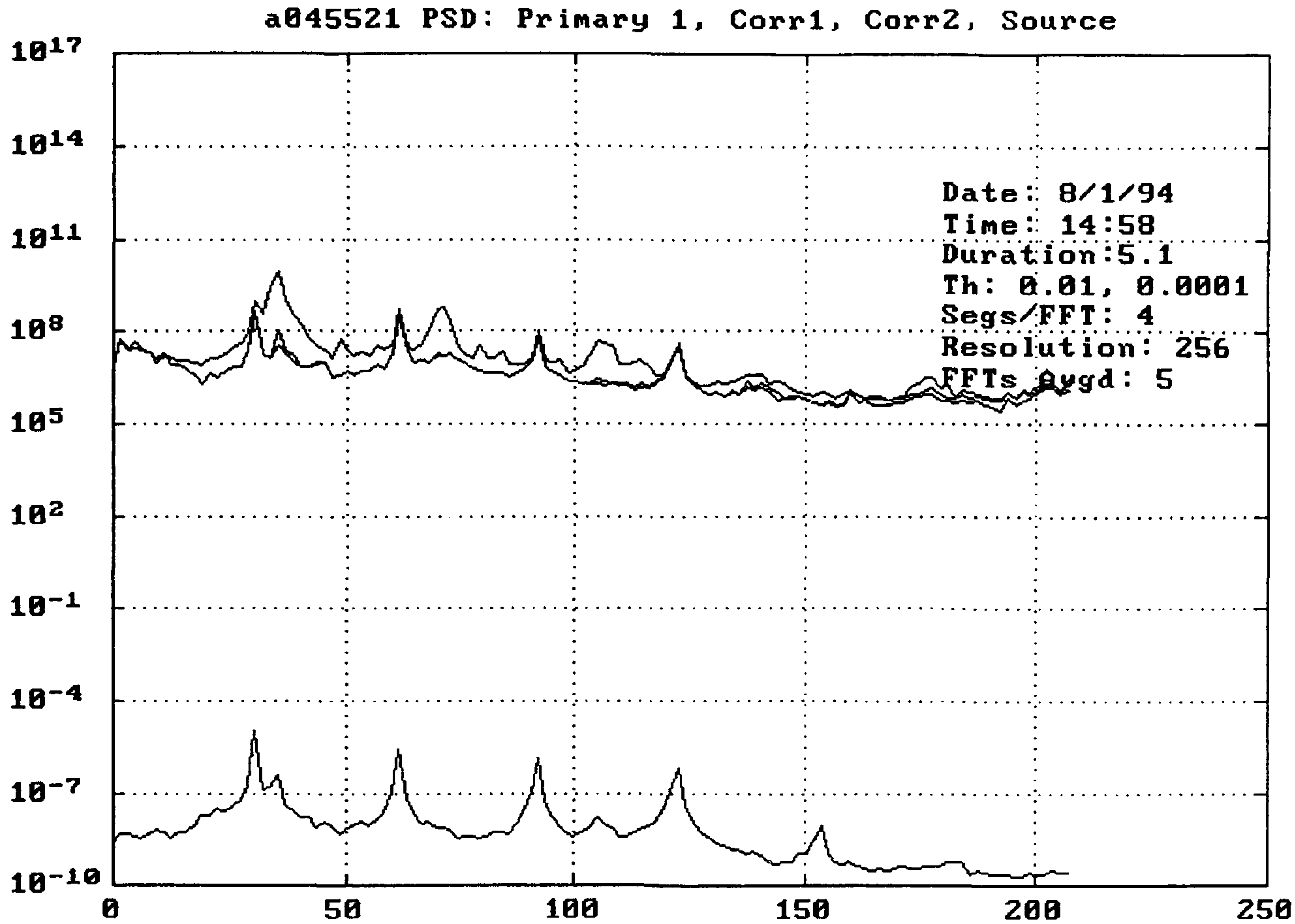


Figure V-2. Same as Figure V-1 but with Uncorrected and the two Corrected curves overlaid for easy visual comparison

The uppermost curve represents the raw (except for low pass filtering) output of the microphone designated as primary sensor 1. The bottom curve shows the source loudspeaker signal as picked up by a microphone placed on top of the speaker. Curves 2 and 3 respectively are two versions of the noise-cancelled output of Primary sensor 1, using eigenvalue cutoff thresholds respectively of .01 (20 dB) and 0.0001 (40 dB) as shown as "Th:" in the identification block of the figure.

Three large harmonics show up respectively at 62, 92, and 123 Hz. The peak at 31 Hz is accompanied by another, smaller, peak at about 35 Hz.

Turning to the uncorrected primary at the top, we see broad bands of engine noise around 65-80 Hz as well as 100-115 Hz. In the corrected curves, this engine noise has been removed while the source peaks remain at substantially their uncorrected level. (The small source peak above 150 Hz appears to never have made it to primary sensor 1.) It is noted that in the corrected output, especially in the second or 40 dB version, the twin peaks at 31-35 Hz have substantially the same shape as in the source, whereas the relative magnitudes of each twin are highly distorted by engine noise in the uncorrected version.

To permit better visual interpretation of the actual noise cancellation and signal retention, the top three curves, uncorrected and two version of corrected, are shown overlaid in Figure V-2. This overlay makes it very clear that the signal has been completely preserved, while a substantial quantity of engine self noise extending over the entire frequency range has been eliminated. In terms of signal to noise improvement, the signal stands out some 15 dB over the noise in the corrected versions, whereas it was submerged in the uncorrected one, suggesting a signal to noise improvement of some 15 dB. This improvement is not atypical for the SFR-SVD Method -- improvements of up to 20 dB and more have been common in past experience. Figure V-2 also demonstrates that the noise cancellation is able to separate signal from noise even when their frequencies coincide, as is the case in the 30-40 Hz range of frequencies. This range clearly contains not only the twin peaks at and near the source fundamental frequency, but it also contains the fundamental range of frequencies of the vehicle engine, and the noise cancellation program clearly has no difficulty in separating the two. Noise some 20 dB in excess of the signal was successfully eliminated while the signal remained completely intact.

In the run shown in Figures V-1 and 2, only four of the seven references were actually applied. This run and its ramifications are discussed at greater length in Section VB.

A. Simulated Test Files (files test0*.psd)

1. Noise Cancellation at frequency resolution 256 (test00.psd)

To verify the correct application of the noise cancellation software to the problem of passive aero-acoustic sensor self interference cancellation, the software was tested using a synthetic sine wave record generated by program testfl1.for. In Figure VA-1, the channels are shown consecutively as 8 "primaries" (curves 1-8), 7 "references" (9-15), and the "source" (16). Each vertical division represents a factor or 10^6 or 60 dB.

Each reference channel (ch. 9 - 15) consisted of a single sine wave at a frequency unique to that sensor. The seven simulated noise frequencies were 7, 11, 20, 36, 50, 74, and 119 Hz respectively. The reason they do not appear as "line" spectra is of course the result of the effectively rectangular window used in the FFT calculations. The source channel (16) likewise was a single sine wave, in this case 170 Hz. The primary channels were chosen as linear combinations of the reference and source signals. None of the primaries except Channels 7 and 8 contained the source frequency, with the result that no significant signal peaks show up in their corrected power spectral densities. Primary Channel 7 (Fig. VA-2), on the other hand , contained equal amplitudes of the two references at 74 Hz and 119 Hz, and the source at 170 Hz. Primary channel 8 (Fig. VA-3) in turn contained equal amplitudes of all 8 frequencies contained in the source and the seven references. Thus, when primary channels 7 and 8 are corrected or noise cancelled as shown in these two figures, each shows only the source frequency peak at 170 Hz. The magnitude of the remaining low-level noise peaks appearing in the corrected primary curves are a function of the thresholds which have been set.

In Primary channel 7, Figure VA-2, the PSD is shown along with two versions of its corrected value, along with the source PSD at the bottom for comparison. Each division on the y-axis corresponds to 30 dB. The uncorrected PSD of primary sensor 7 contains peaks at 74, 119, and 170 Hz as expected from what was put into channel 7. The second curve is Primary 7 when corrected to a 20 dB threshold level on the eigenvalues. The peaks at 74 and 119 Hz have been effectively eliminated, although the lower frequencies

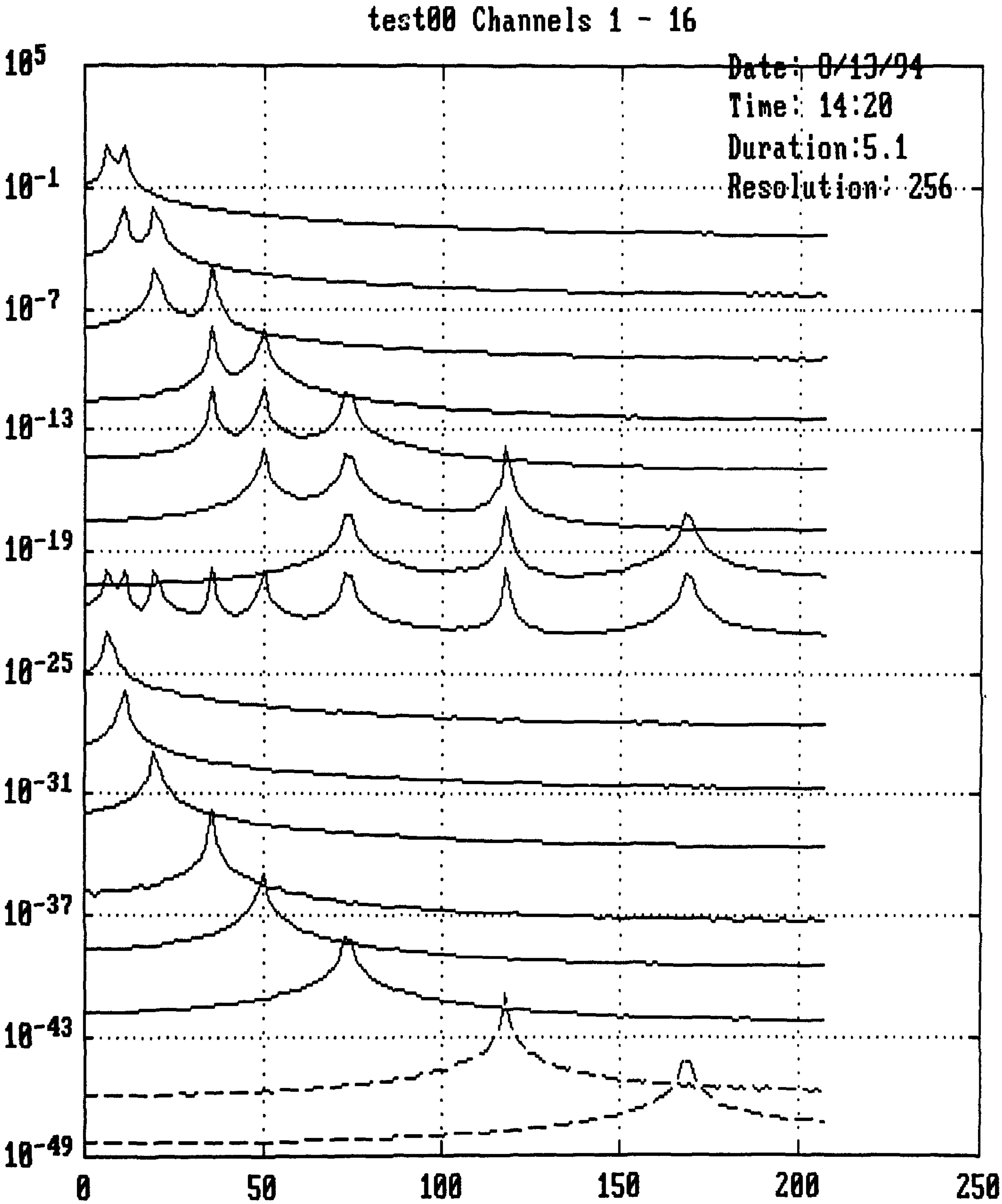


Figure VA-1. PSD of all 16 channels: simulated test

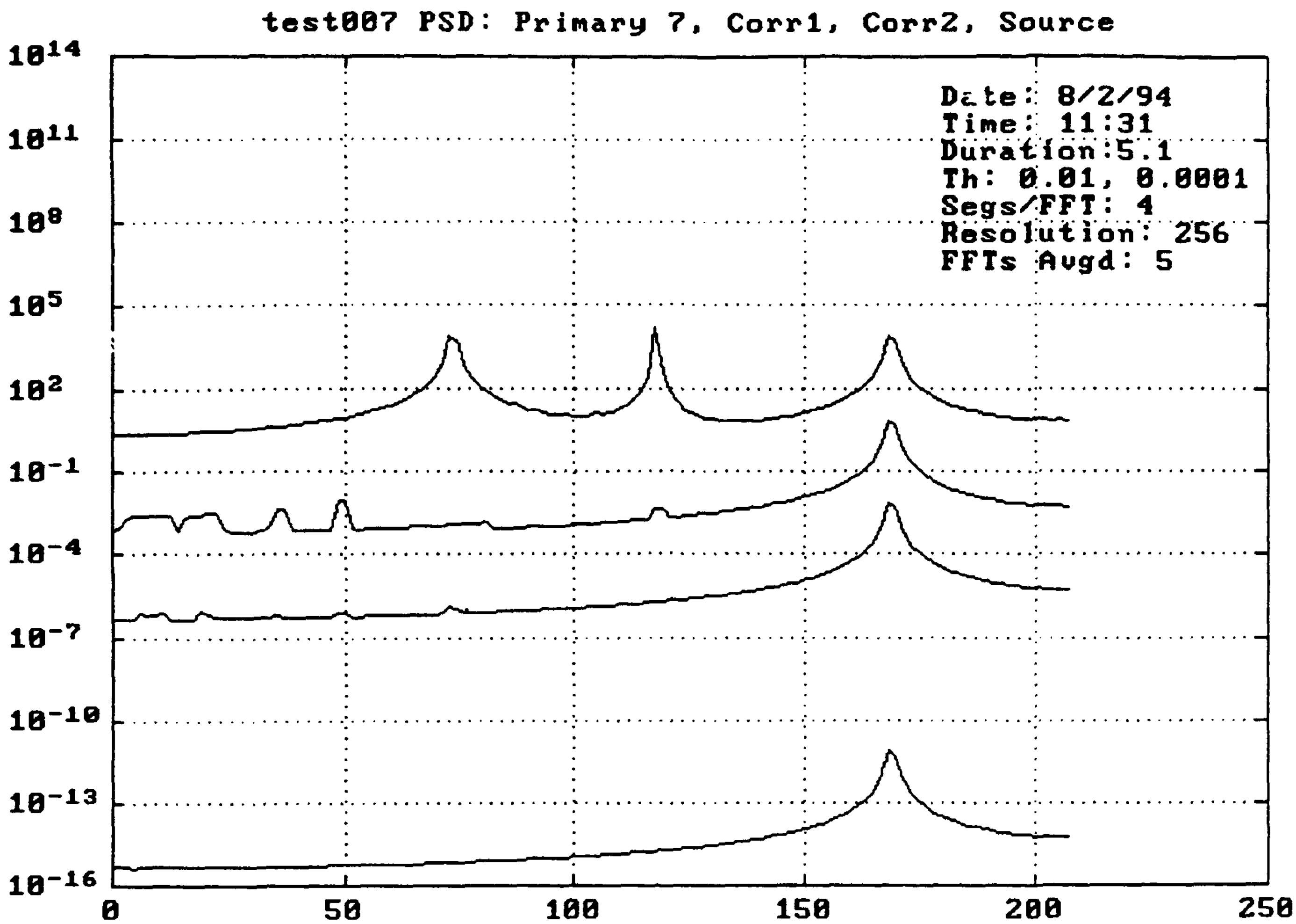


Figure VA-2. PSD's, Primary 7, corrected (2), source: simulated test

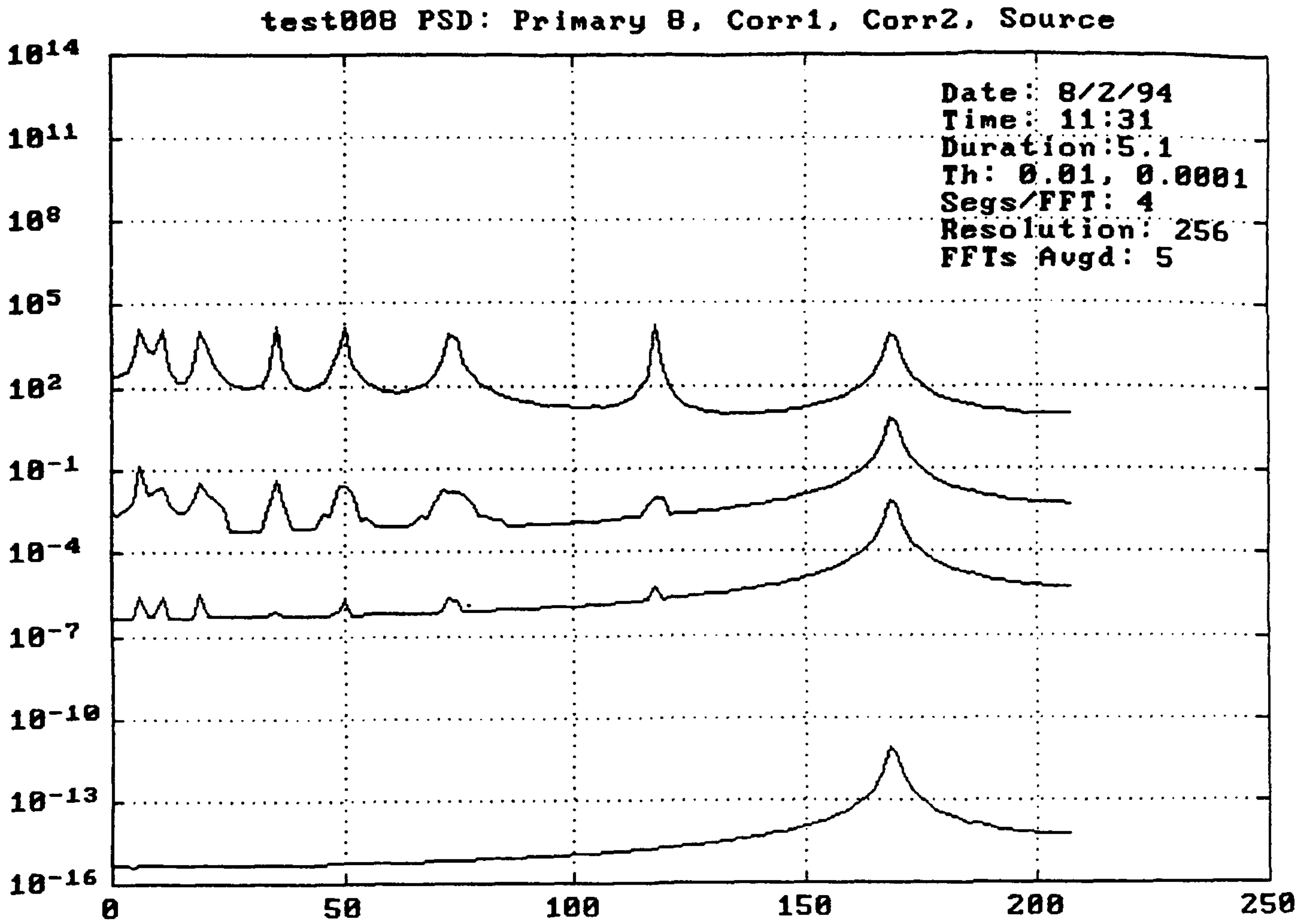


Figure VA-3. PSD's, Primary 8, corrected (2), source: simulated test

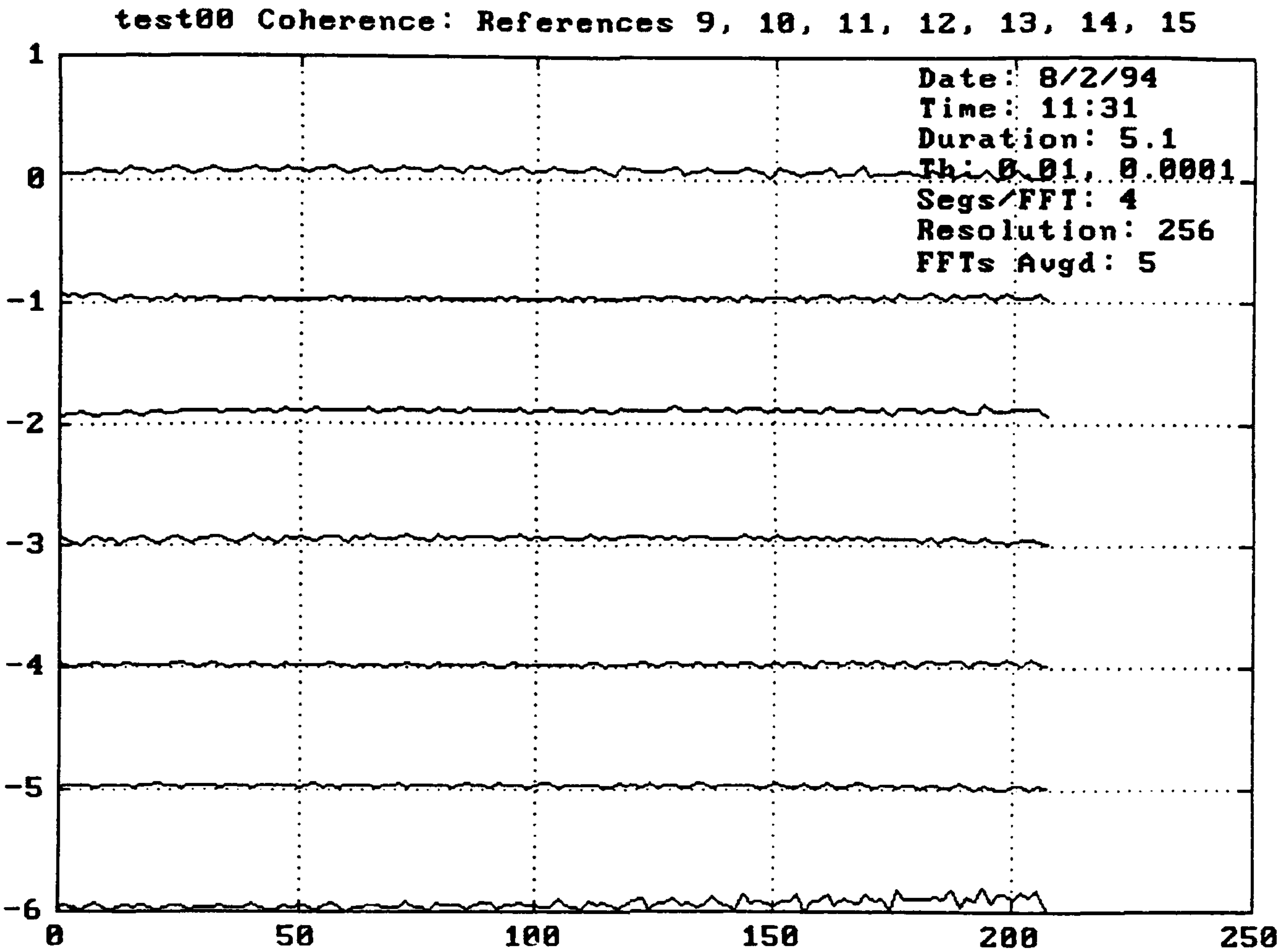


Figure VA-4. Coherences of references with source: simulated test. (Note: Since the simulated test constituted deterministic sine waves, not stochastic functions, these coherences have no physical significance.)

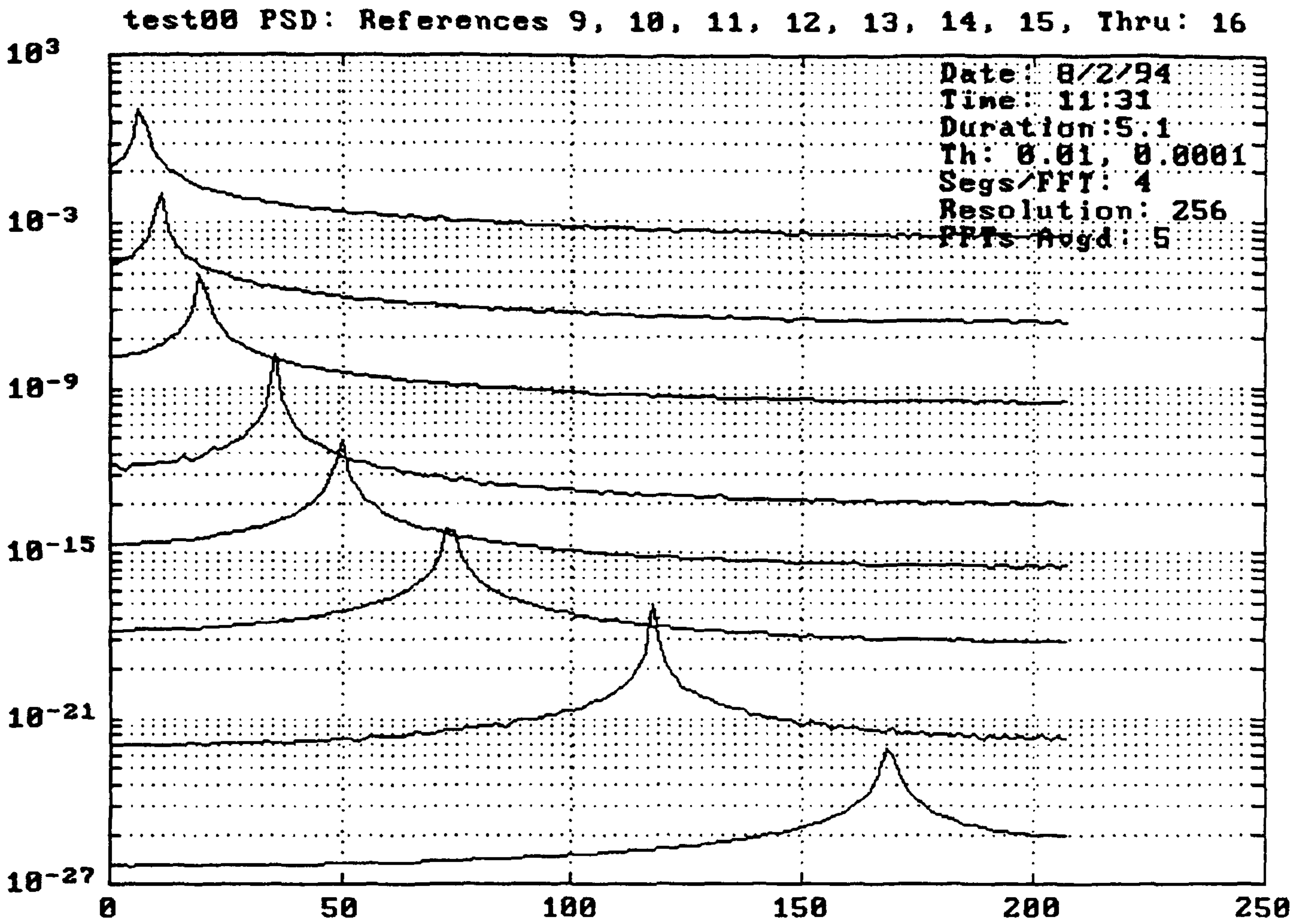


Figure VA-5. PSD's, References and source: simulated test

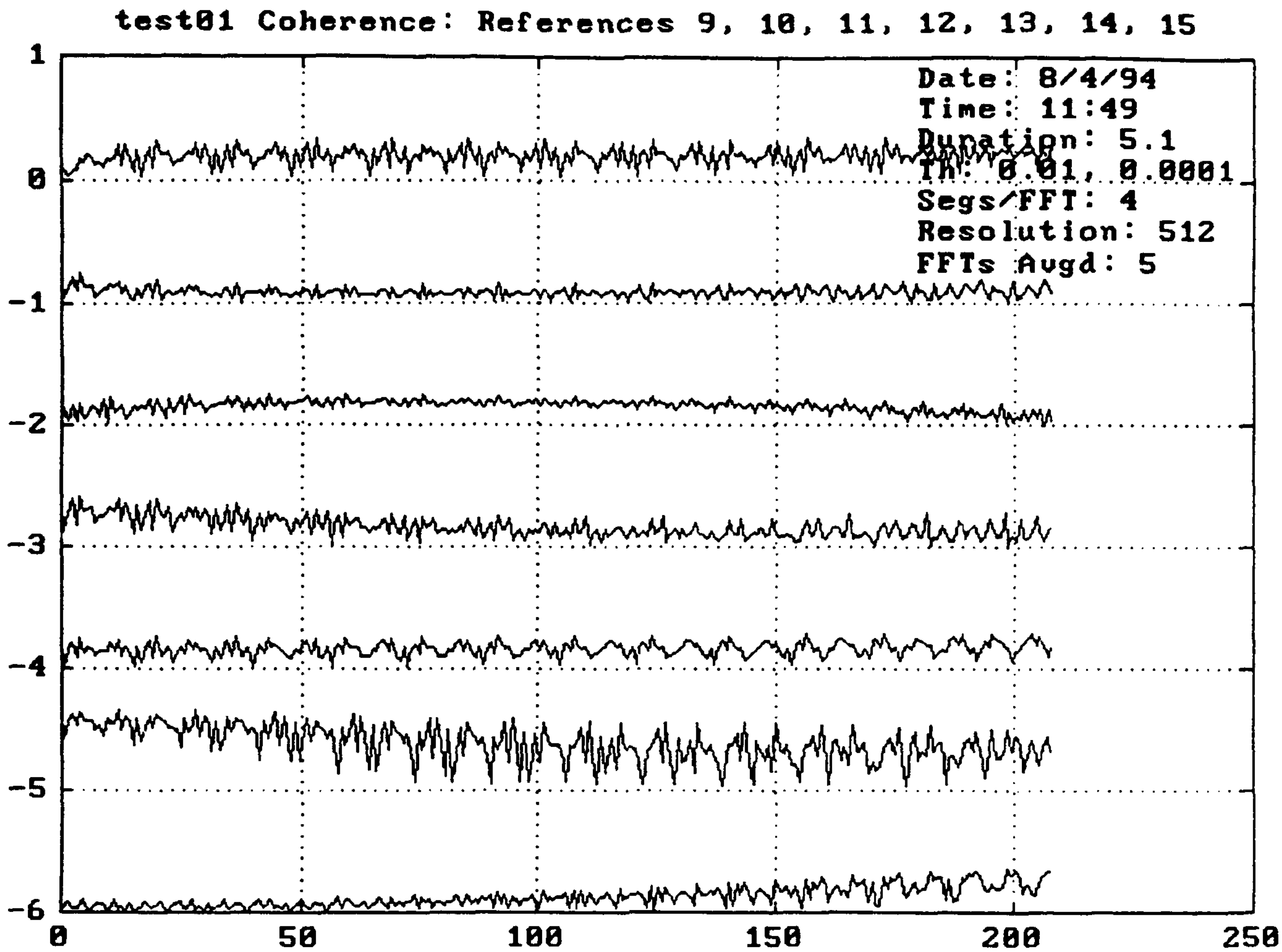


Figure VA-6. Coherences as in Fig. VA-1, but with 512 resolution: simulated test; no physical significance

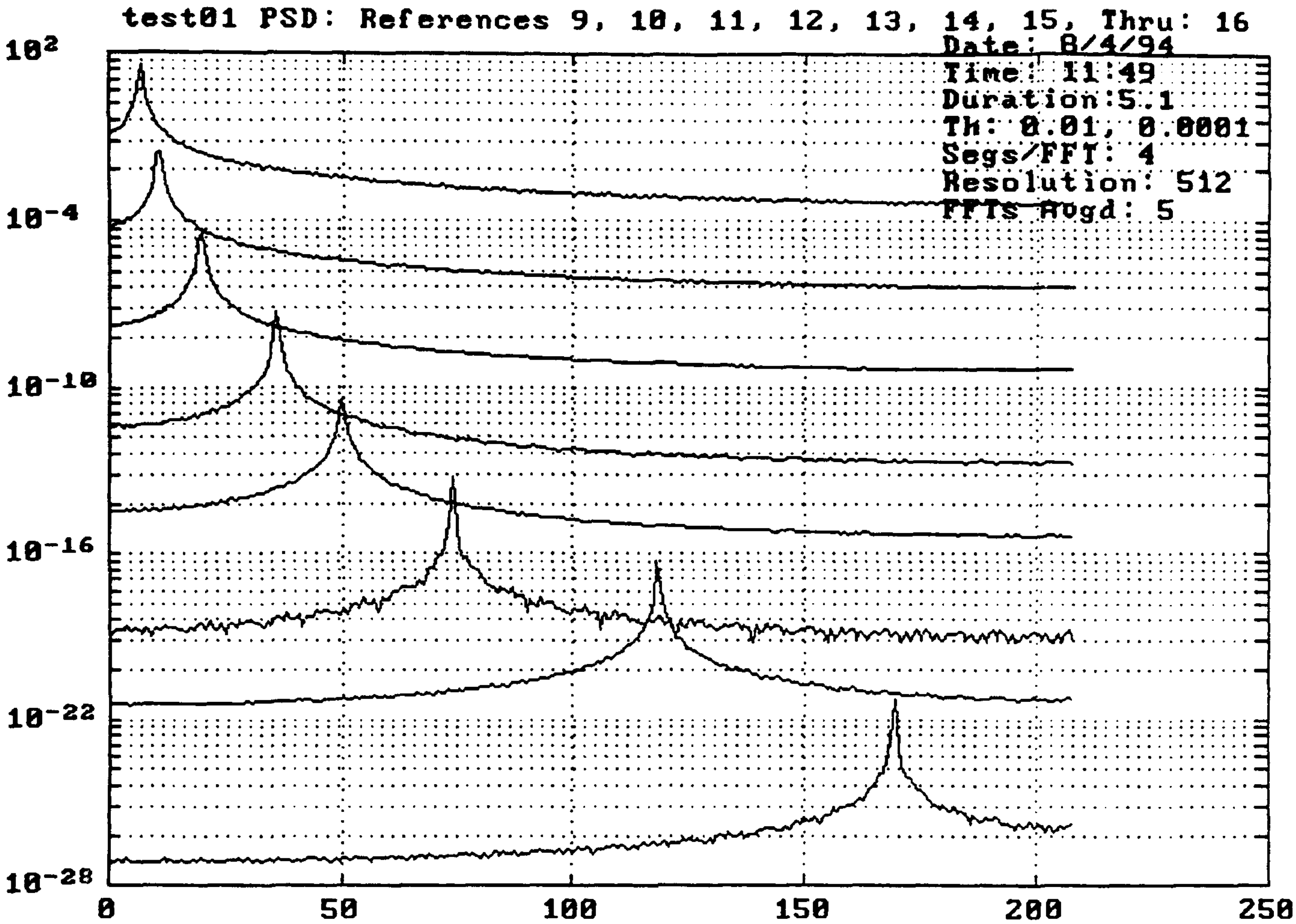


Figure VA-7. PSD's, References and source as in Fig. VA-3, but with 512 resolution: simulated test

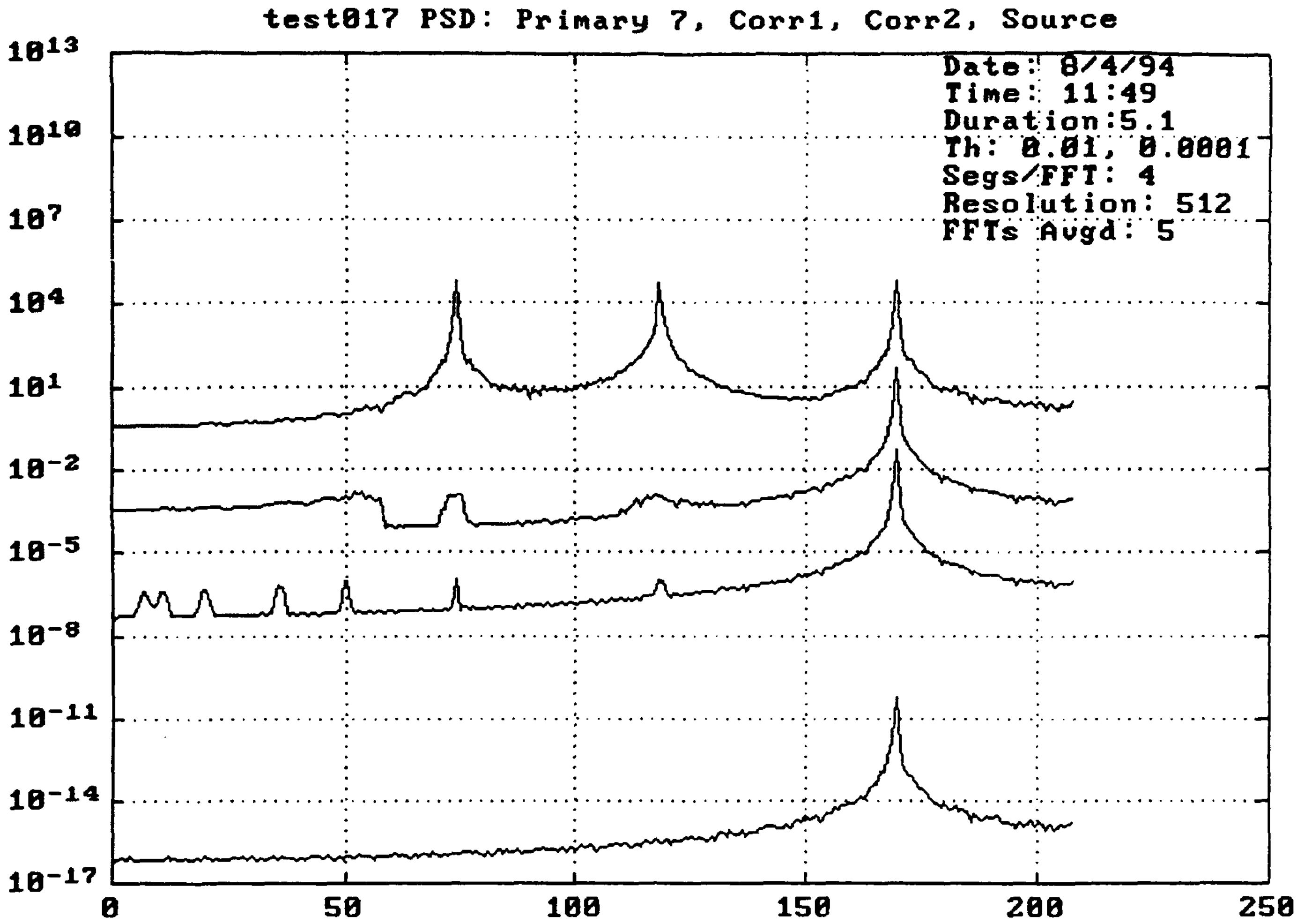


Figure VA-8. PSD's, Primary 7, corrected (2), source as in Fig. VA-4, but with 512 resolution: simulated test

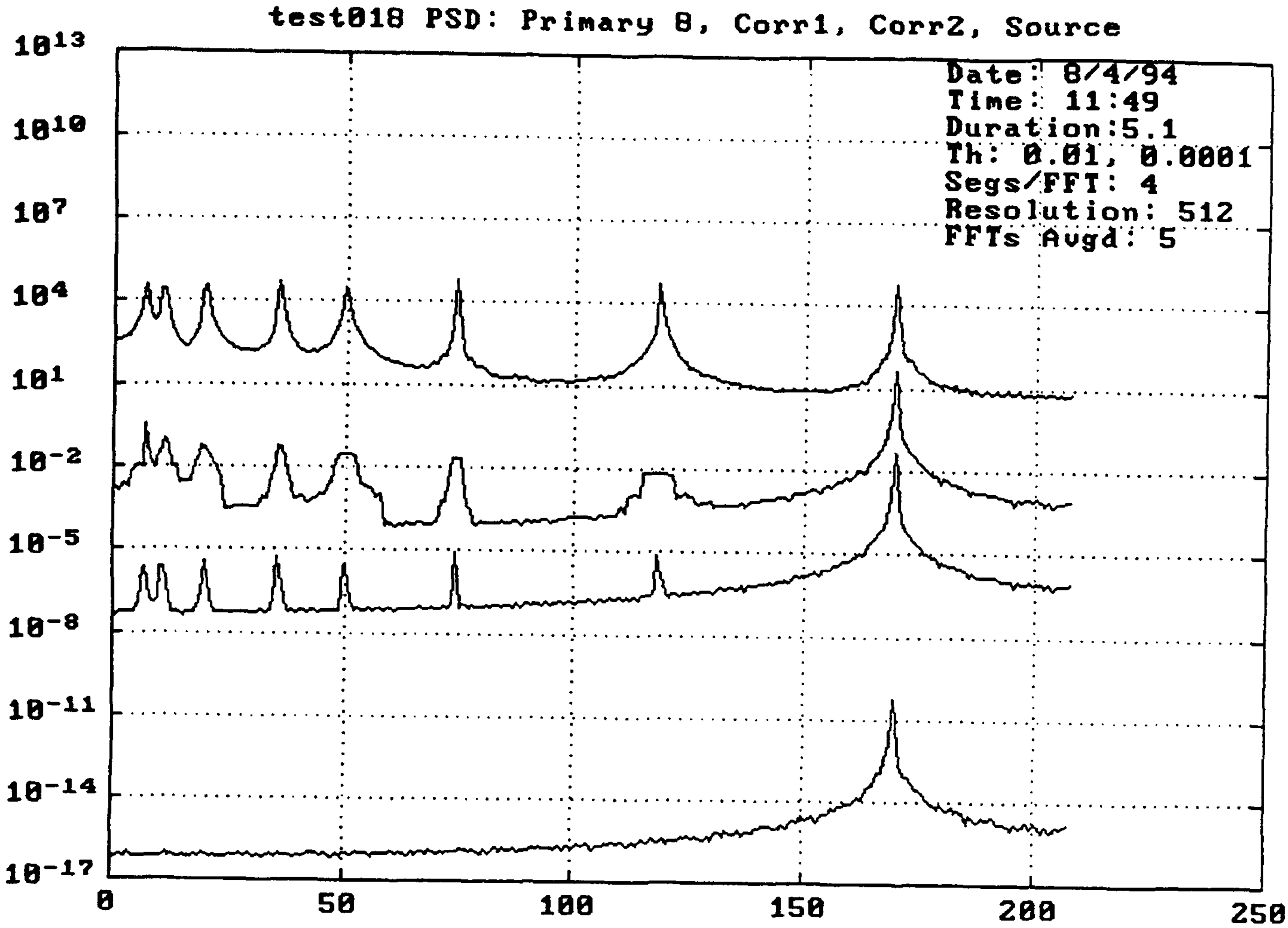


Figure VA-9. PSD's, Primary 8, corrected (2), source as in Fig. VA-5, but with 512 resolution: simulated test

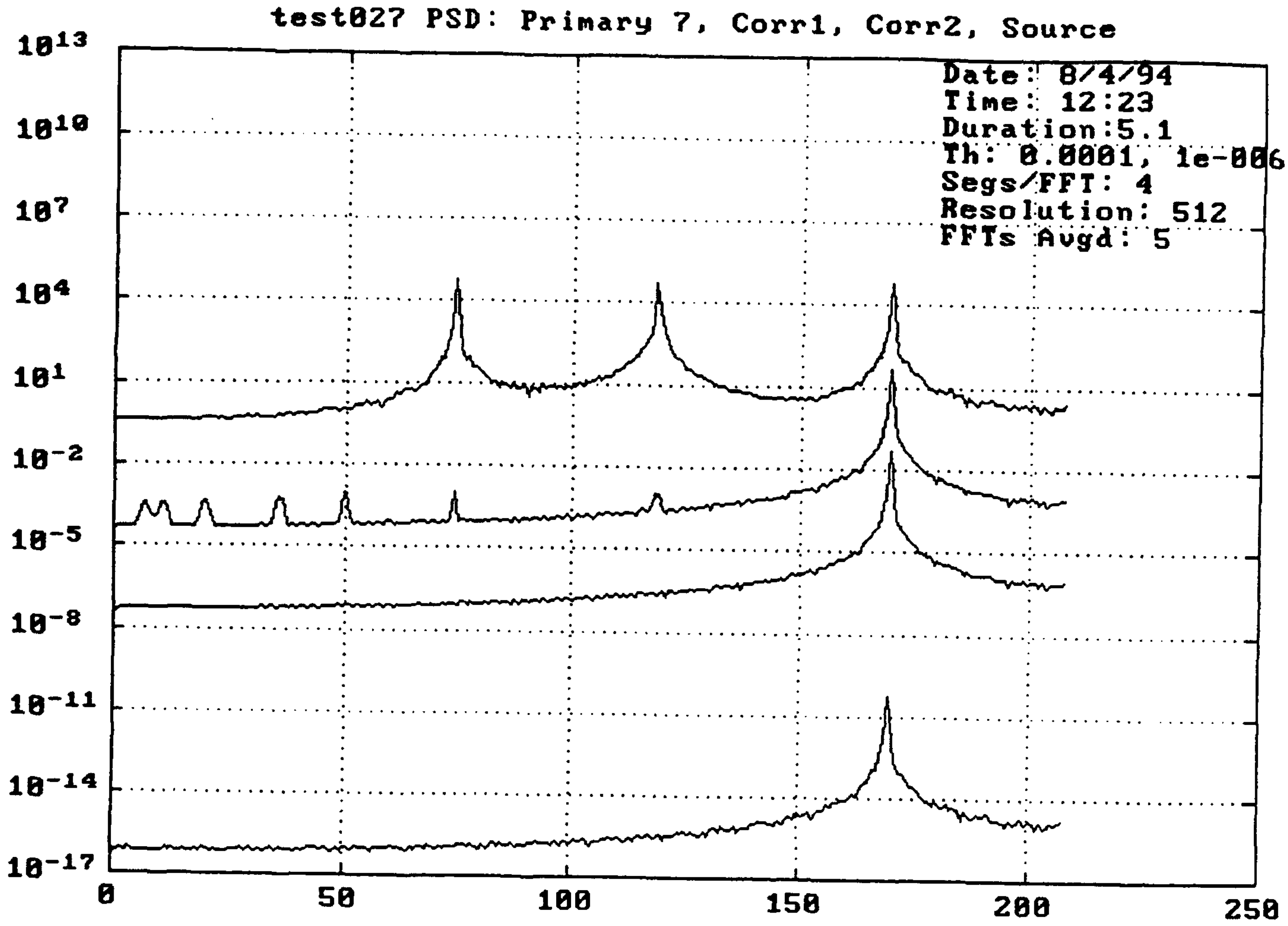


Figure VA-10. PSD's, Primary 7, corrected (2), source as in Fig. VA-8, but with 512 resolution: simulated test

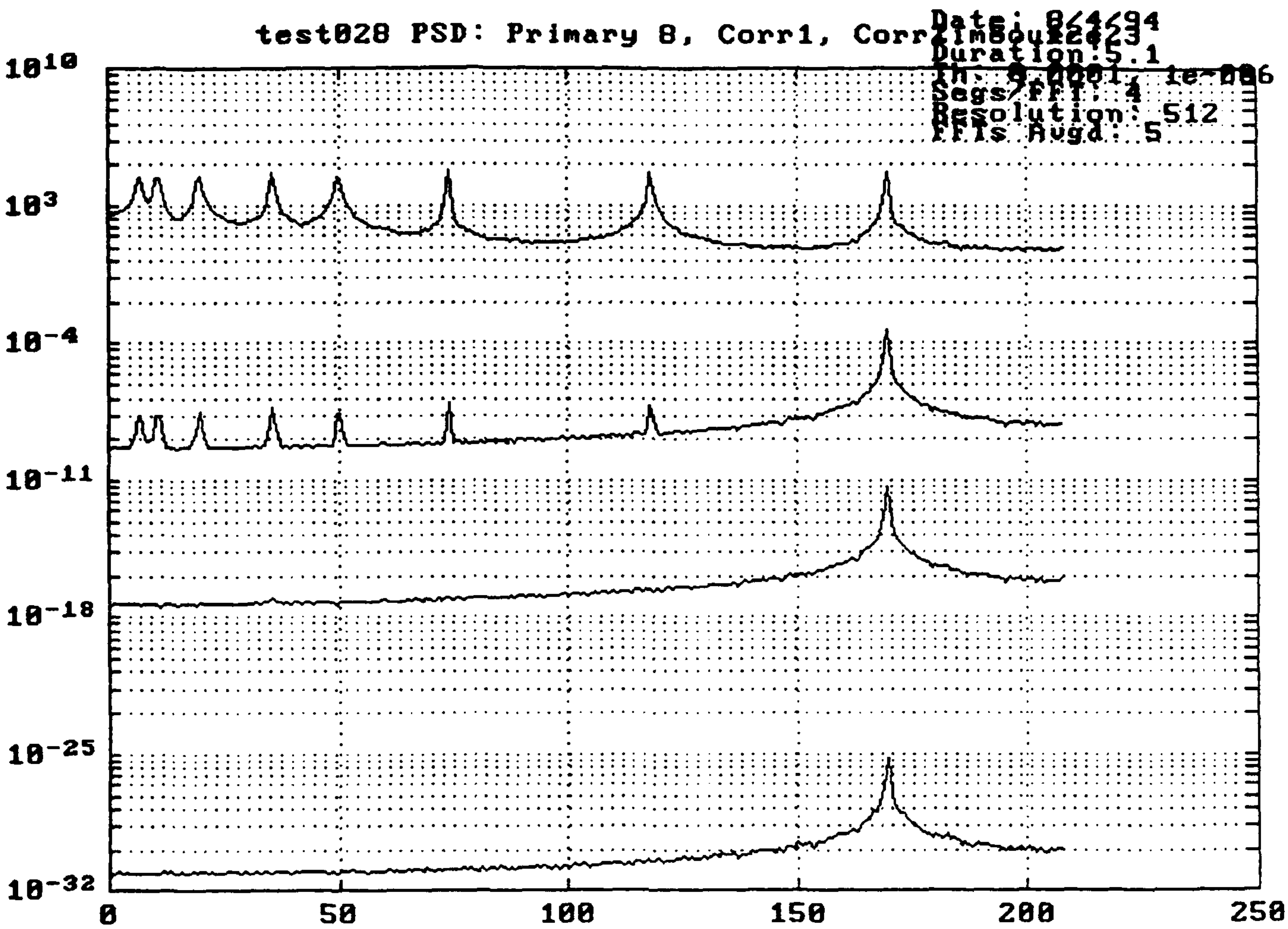


Figure VA-11. PSD's, Primary 8, corrected (2), source as in Fig. VA-9, but with 512 resolution: simulated test

do appear some 30 dB below the source peak at 170 Hz. When the eigenvalue threshold is set to 40 dB, the resulting corrected curve appears remarkably like the source signal.

Figure VA-3 shows the PSD of primary sensor 8, its two corrected values, and against the source. In this case as expected, the uncorrected primary shows all eight frequency peaks as put there. The 20 dB threshold corrected PSD lifts the source channel at 170 hz some 20 dB above the highest "noise" frequency at 7 Hz, while the 40 dB correction provides some 40 dB of protection against the highest noise level at 119 Hz. Again, the lower threshold value eliminates more of the remnants of "noise" frequencies in the primary.

Figure VA-4 shows the coherence at resolution 256, of each reference with the source. Each of the 7 curves is plotted to a vertical scale from 0 to 1; the labels on the y-axis have no other meaning. These coherences are all close to zero over the frequency band of interest, suggesting that no substantial source energy entered the references. However, it should be kept firmly in mind that for this particular example, the "coherences" are based on deterministic time functions (sine waves), and therefore really have little if any meaning. We would have been neither disappointed nor delighted had the plots come out at something other than close to zero.

Figure VA-5 depicts the power spectral density (PSD) for each of the seven references (ch. 9 - 16), and at the bottom, the source channel (16). The main subdivisions on the y-axis are separated by a factor of 10^6 or 60 dB; the absolute values on the y-scale, as well as the minor divisions, also have no significance.

2. Cancellation at resolution 512 and the relationship between resolution and optimal thresholds (test02.psd, test03.psd)

One of the noise cancellation parameters that has to be chosen is the desired resolution in the frequency domain. This resolution, which we have normally set at 256, is the number of source file time points used for an FFT. At resolution 256, the Nyquist frequency occurs at the 129th value in the frequency domain, so that the frequency range of interest is divided into 129 bins. Since the highest, or Nyquist, frequency is at half the decimated sampling rate (2083/5/2 or 208 Hz, the width of a frequency bin for resolution 256 would be 208/129 or 1.6 Hz. Resolutions of 512 (0.8 Hz) and 1024 (0.4 Hz) were also examined but seemed to offer little additional information and were more difficult to interpret visually.

Figures VA-6 through 13 present an exploration the effect of frequency resolution. Figure VA-6 shows the coherences of the 7 references with the source. These coherences came out larger than when using resolution 256, but again, not too much heed should be paid to their values in the case of this deterministic problem. Figure VA-7 shows the reference power spectra, which are similar to those corresponding to 256 resolution, except for the added jitter. The noise cancelling results for sensors 7 (Fig. VA-8) and 8 (Fig. VA-9) indicate that the thresholds of .01 and .0001 are not very suitable for this higher resolution. Accordingly, the run was repeated with each threshold moved down some 20 dB. The coherences and reference PSD's are of course independent of threshold and therefore identical to to those of Figures VA-6 and 7 respectively, but the noise cancelling performance is improved as shown in Figures VA-10 and VA-11. This demonstrates the inherent interdependence between effective resolutions and effective eigenvalue thresholds. This interaction between resolution and optimal thresholds has been encountered before, and may be thought of as follows: When the bandwidth of the frequency bins is narrower, it is necessary to dig down relatively deeper into the set of noise eigenvalues in order to obtain all those required for successful noise cancellation.

B. Sine Wave Input: 31 Hz source signal (a04555.psd, a04552.psd)

We turn now to a closer examination of the successful example already cited at the beginning of this Section V, when a 31 Hz sine wave was played by the loudspeaker. (In the test log, this source is reported as 25 Hz; the difference results from the fact that the source was not calibrated, and this difference does not affect the validity of the analysis.) But why, in particular, did the use of references have to be restricted to only 4 of the 7 available ones?

We begin with Figure VB-1, which shows all 16 channels recorded for this example: 8 primaries, 7 references, and the source. There are significant source harmonics at 62, 92, 123, and a small one at 154 Hz, which did not make it in a substantial way into any sensor, primary or reference. Figure VB-2 shows an attempted noise cancellation using all 7 references with primary sensor 3 arbitrarily chosen for this study. The result is near total cancellation of the source signal, regardless of which threshold was applied. Why?

Figure VB-3 depicts the power spectral densities of the seven references and the source. The second, third, and fourth (References 10, 11, and 12 – the non-tailpipe microphones)

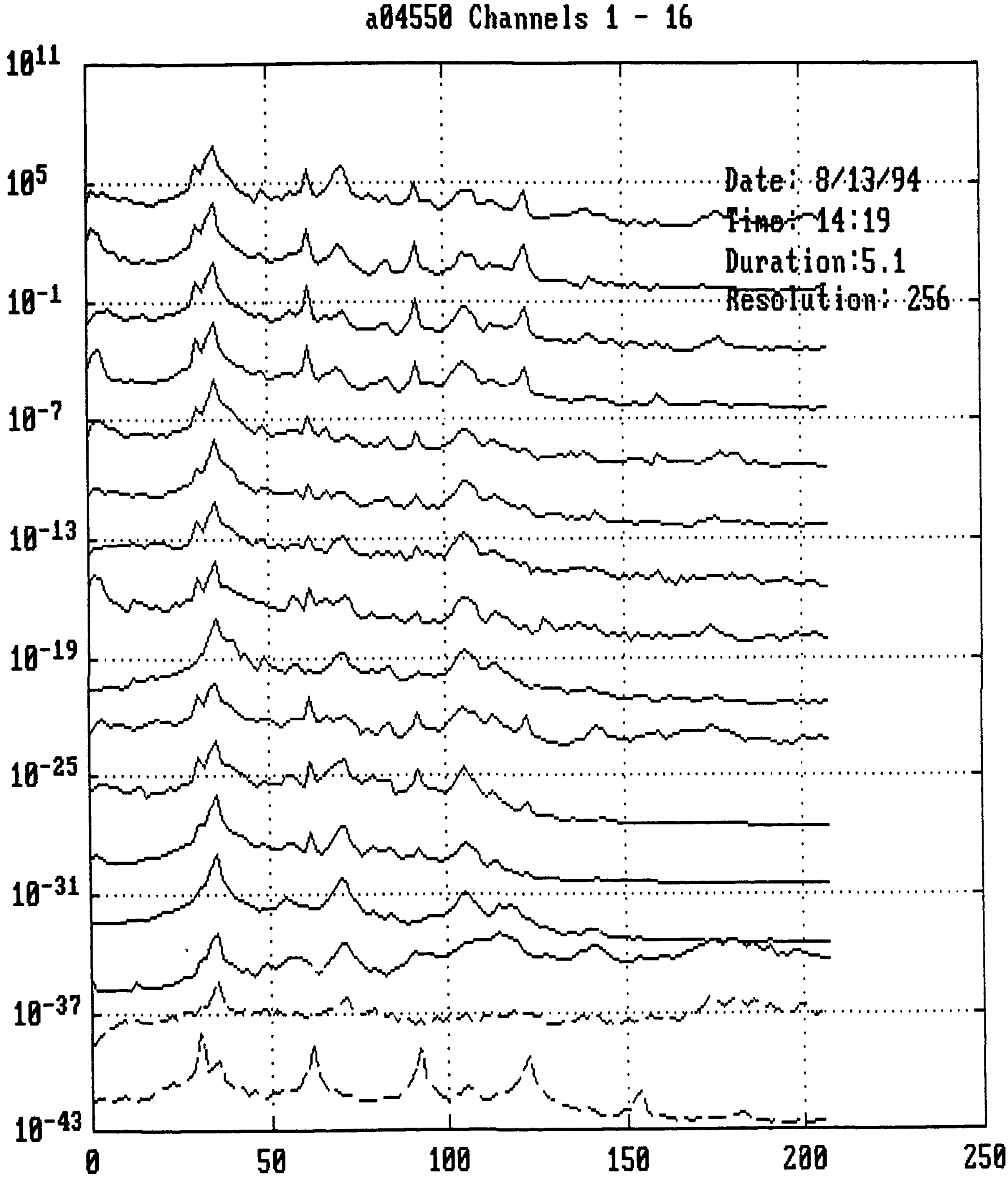


Figure VB-1. PSD of all 16 channels: 25 Hz sine wave

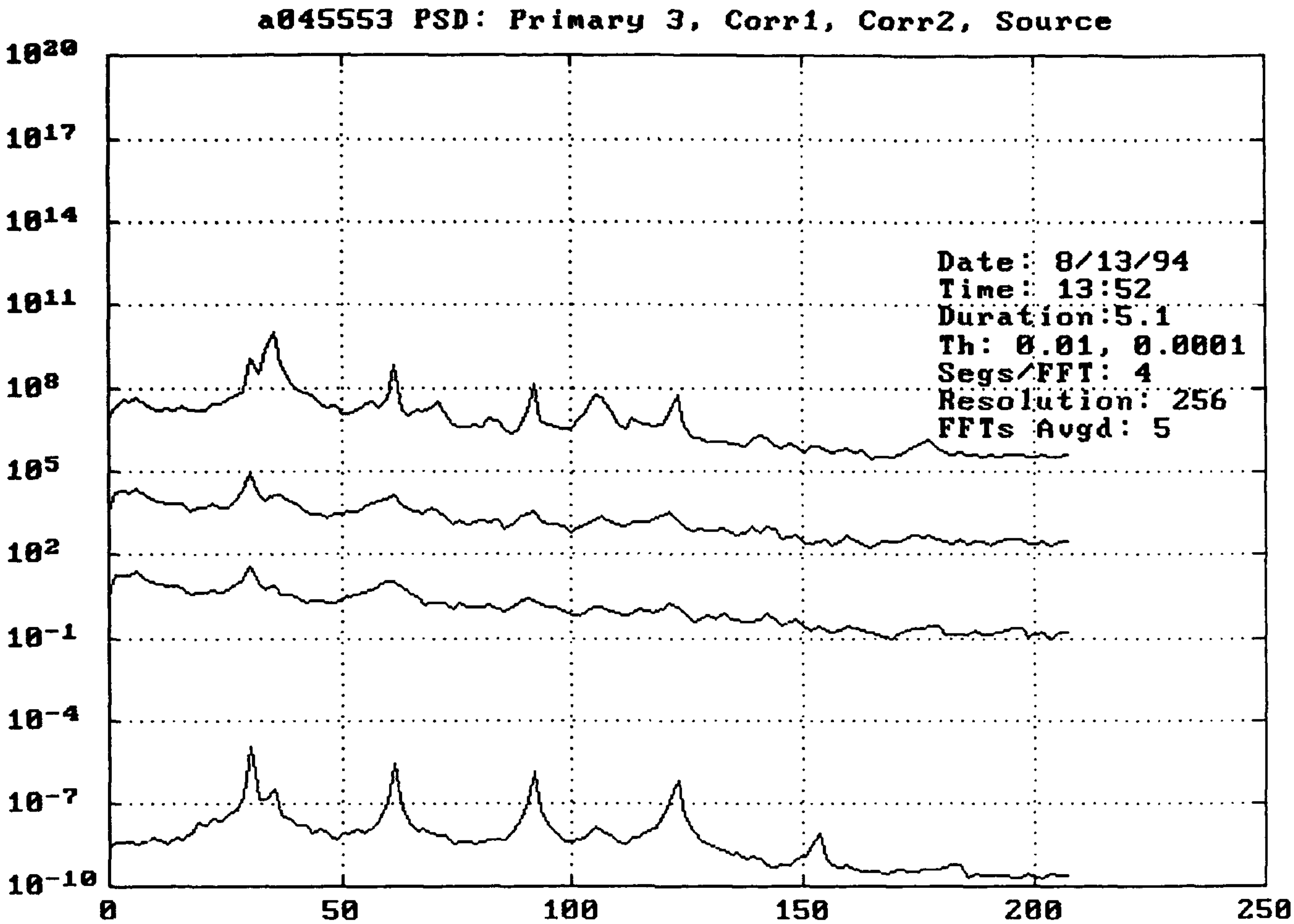


Figure VB-2. PSD's, primary 3, corrected (2), source: 25 Hz sine wave

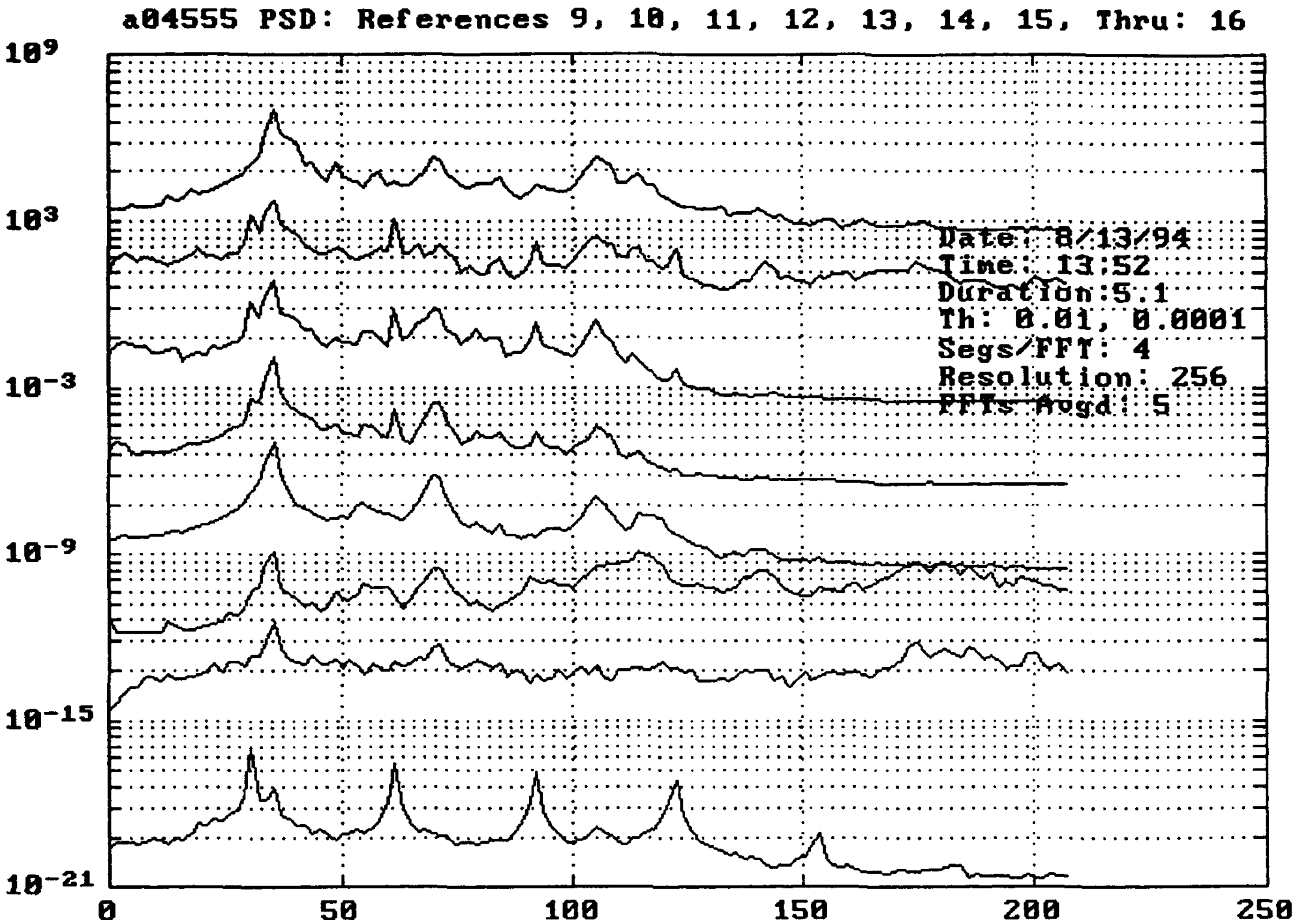


Figure VB-3. PSD's, References and source: 25 Hz sine wave

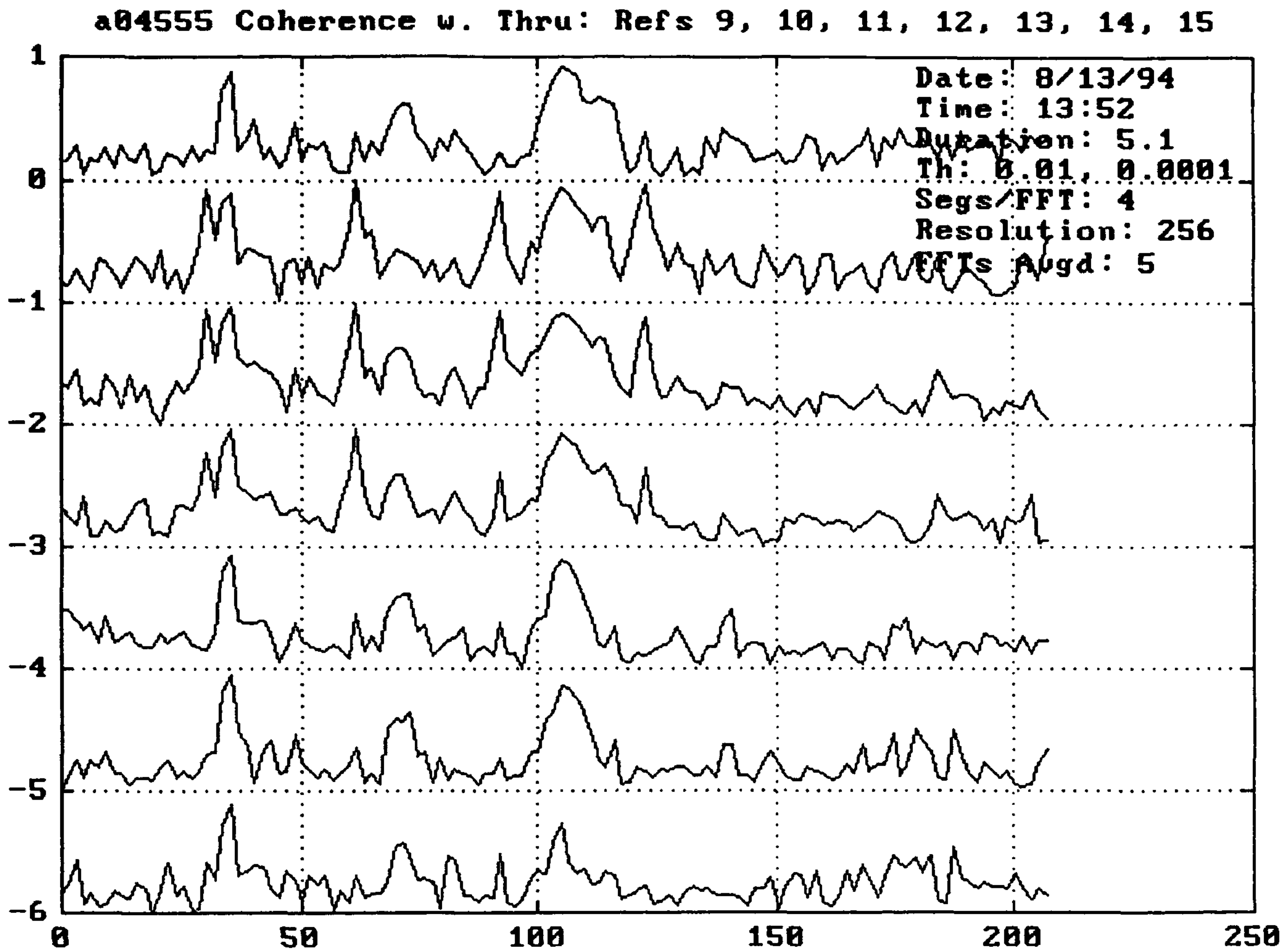


Figure VB-4. Coherences of references with source: 25 Hz sine wave

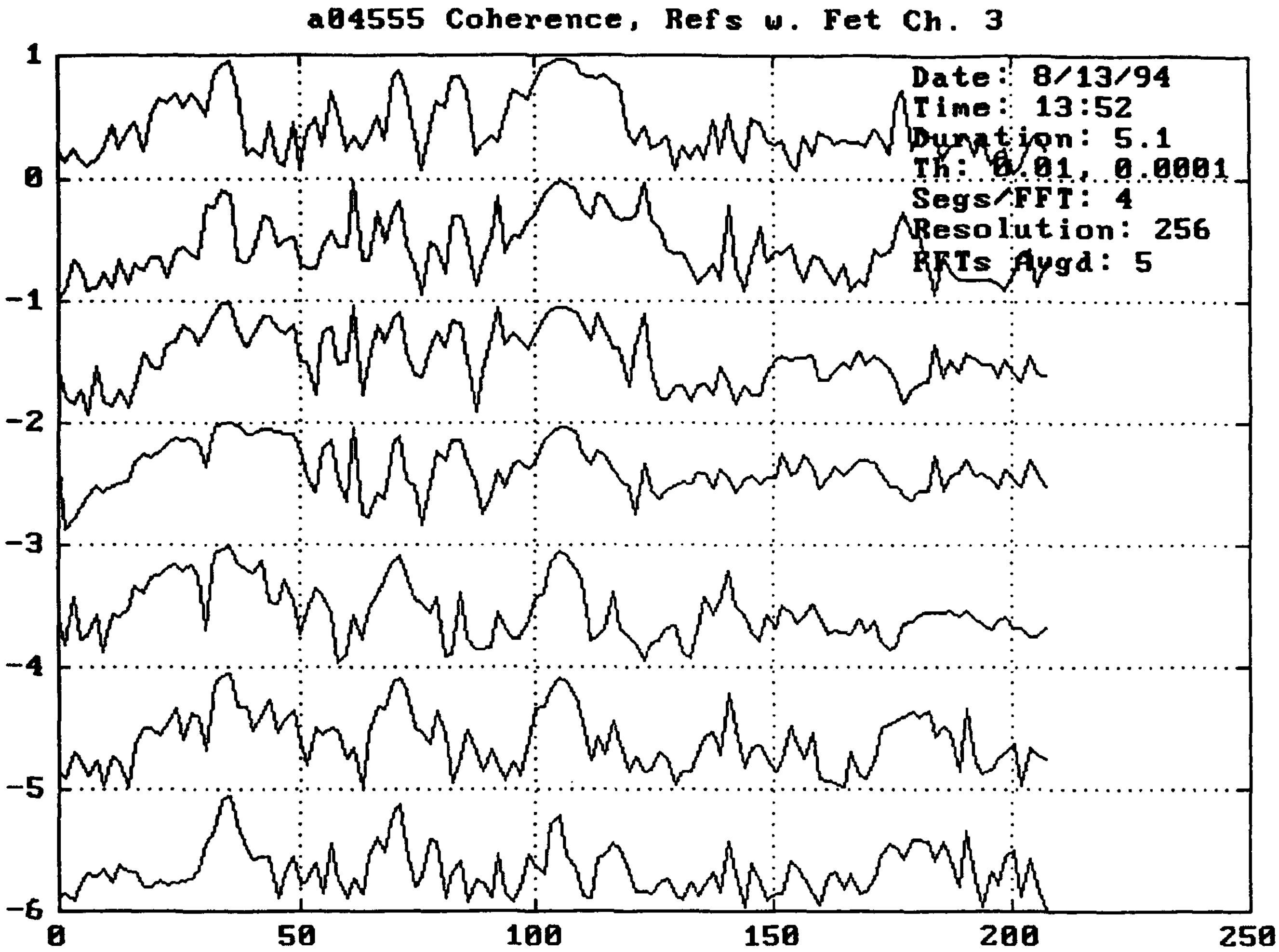


Figure VB-5. Coherences of references with primary 3: 25 Hz sine wave

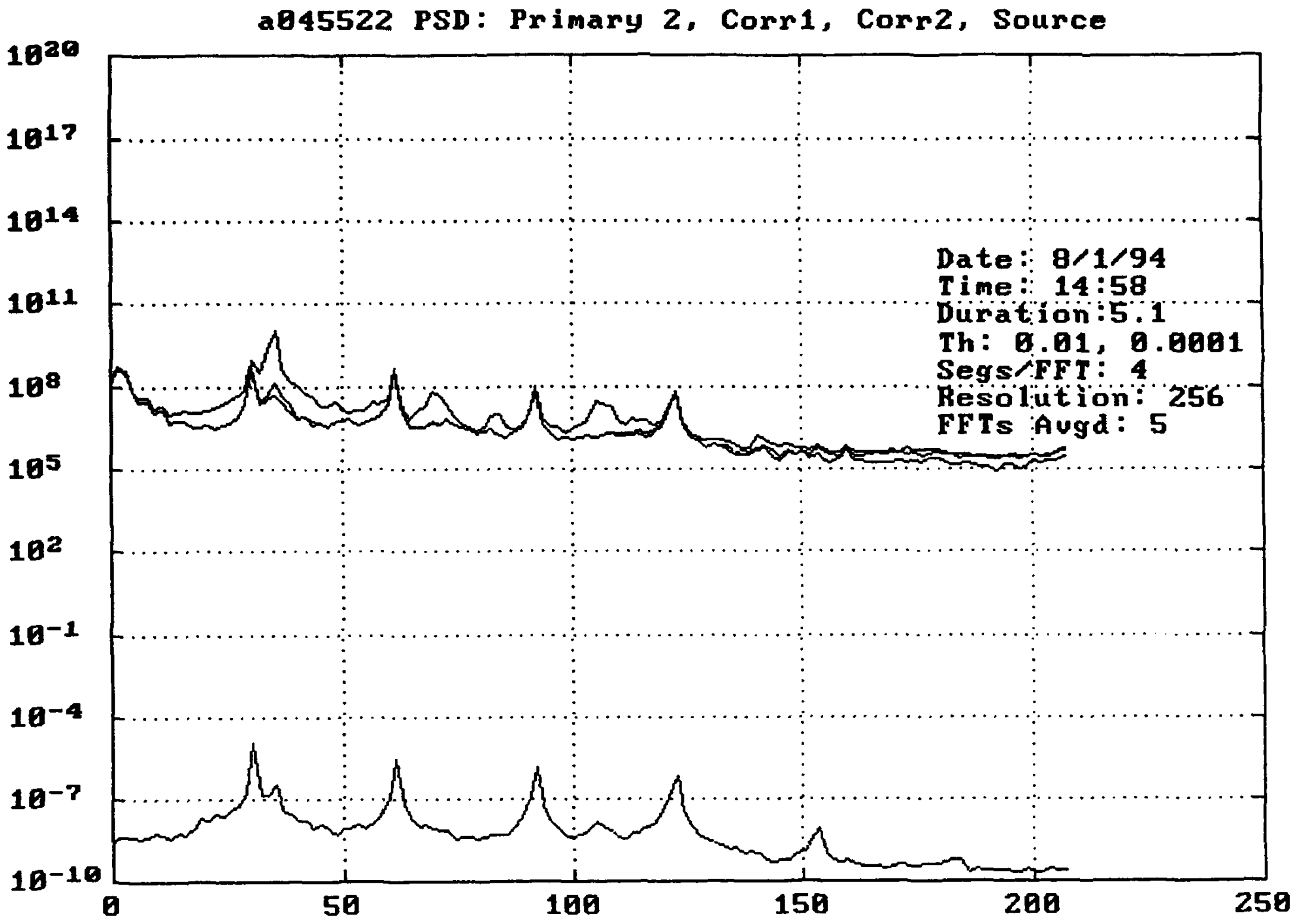


Figure VB-6. PSD's, Primary 2, corrected (2), source: 25 Hz sine wave

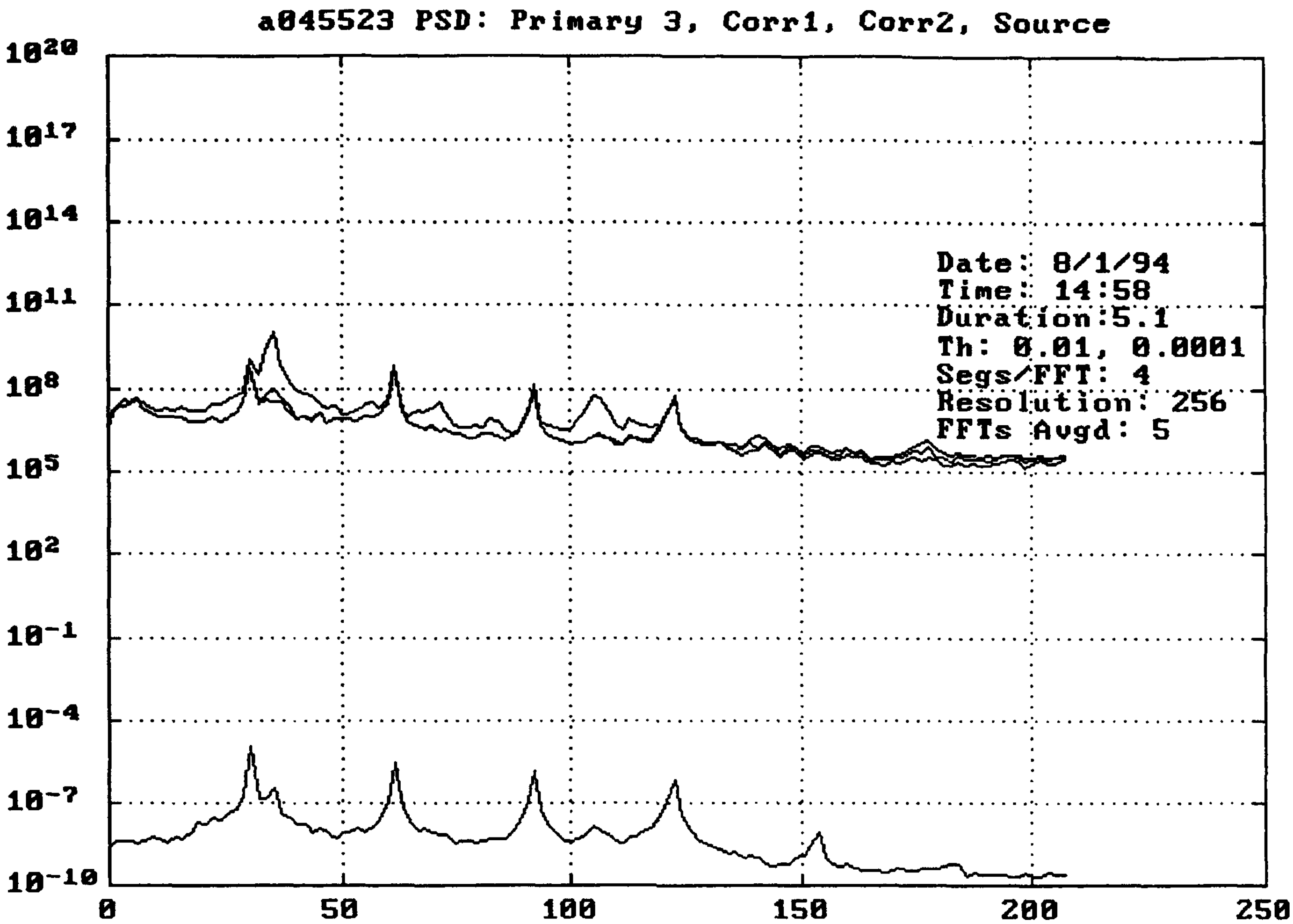


Figure VB-7. PSD's, Primary 3, corrected (2), source: 25 Hz sine wave

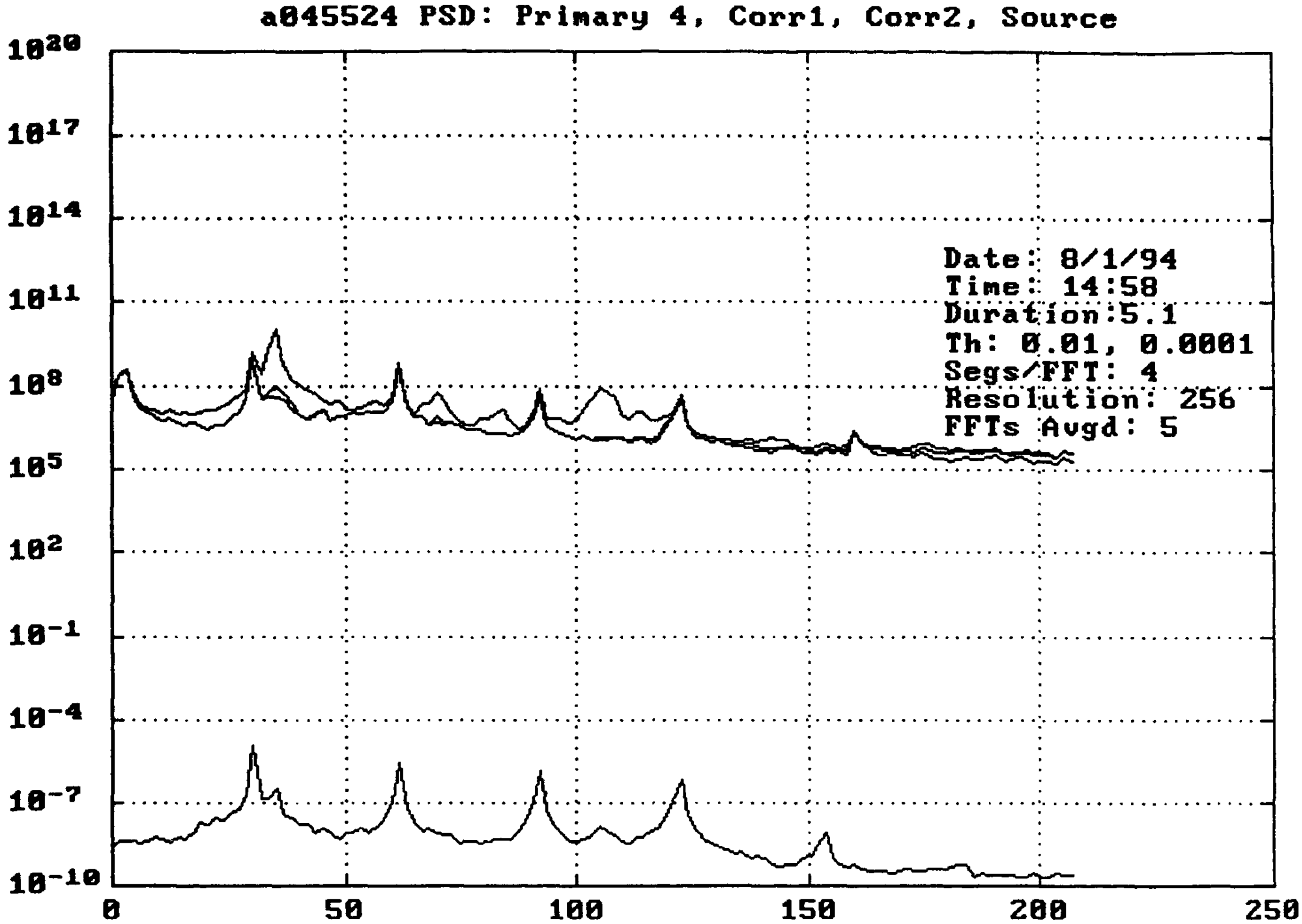


Figure VB-8. PSD's, Primary 4, corrected (2), source: 25 Hz sine wave

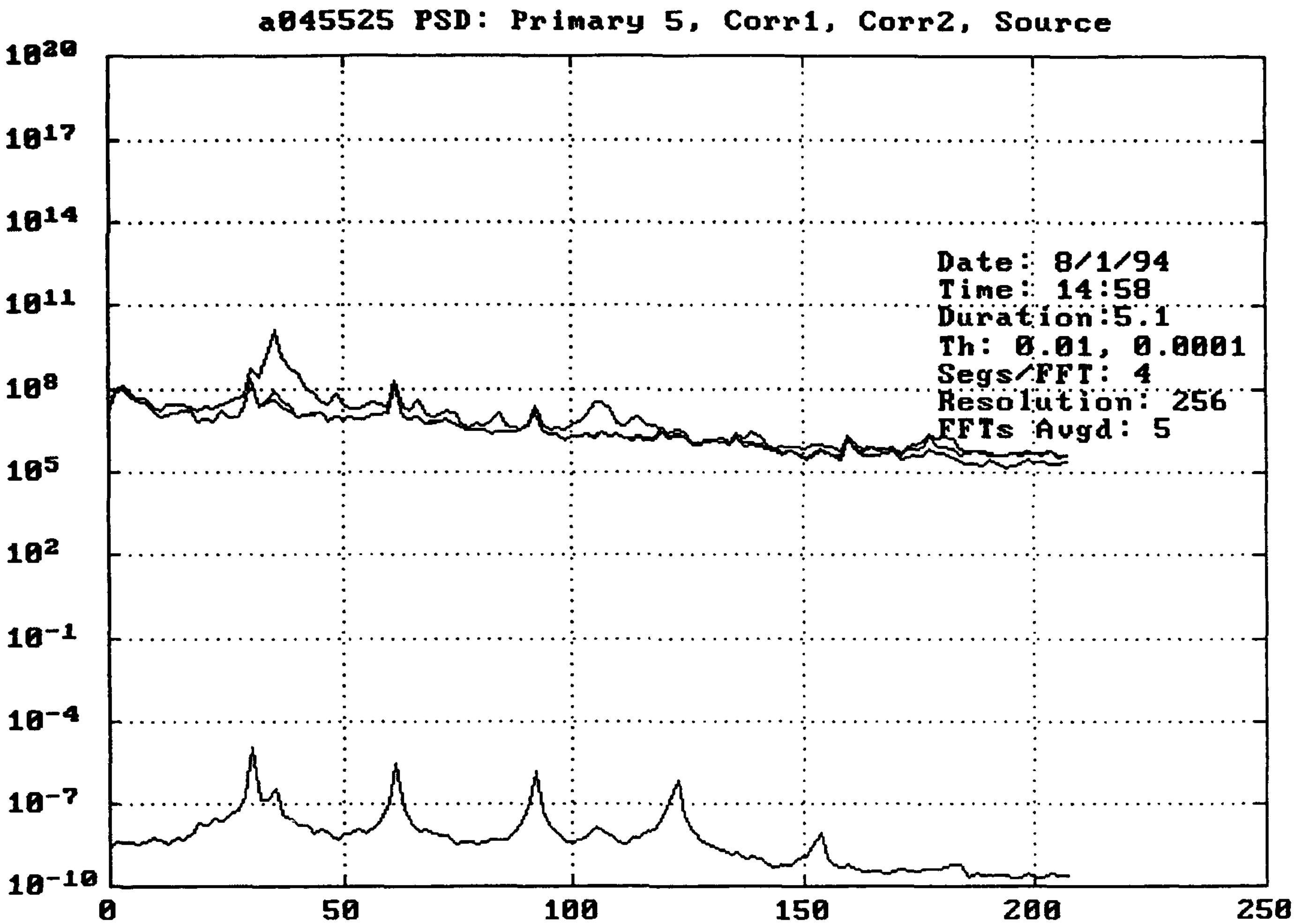


Figure VB-9. PSD's, Primary 5, corrected (2), source: 25 Hz sine wave

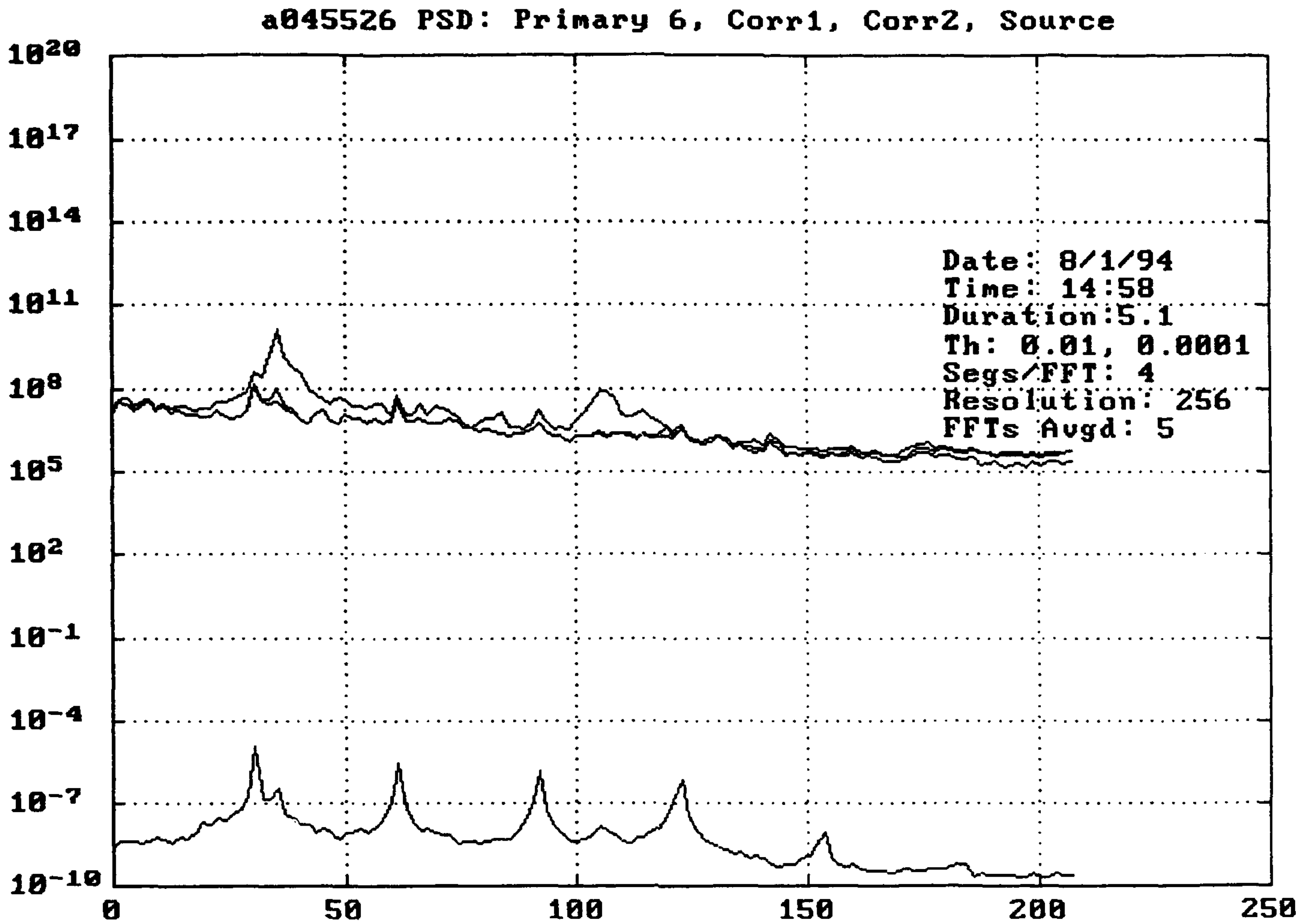


Figure VB-10. PSD's, Primary 6, corrected (2), source: 25 Hz sine wave

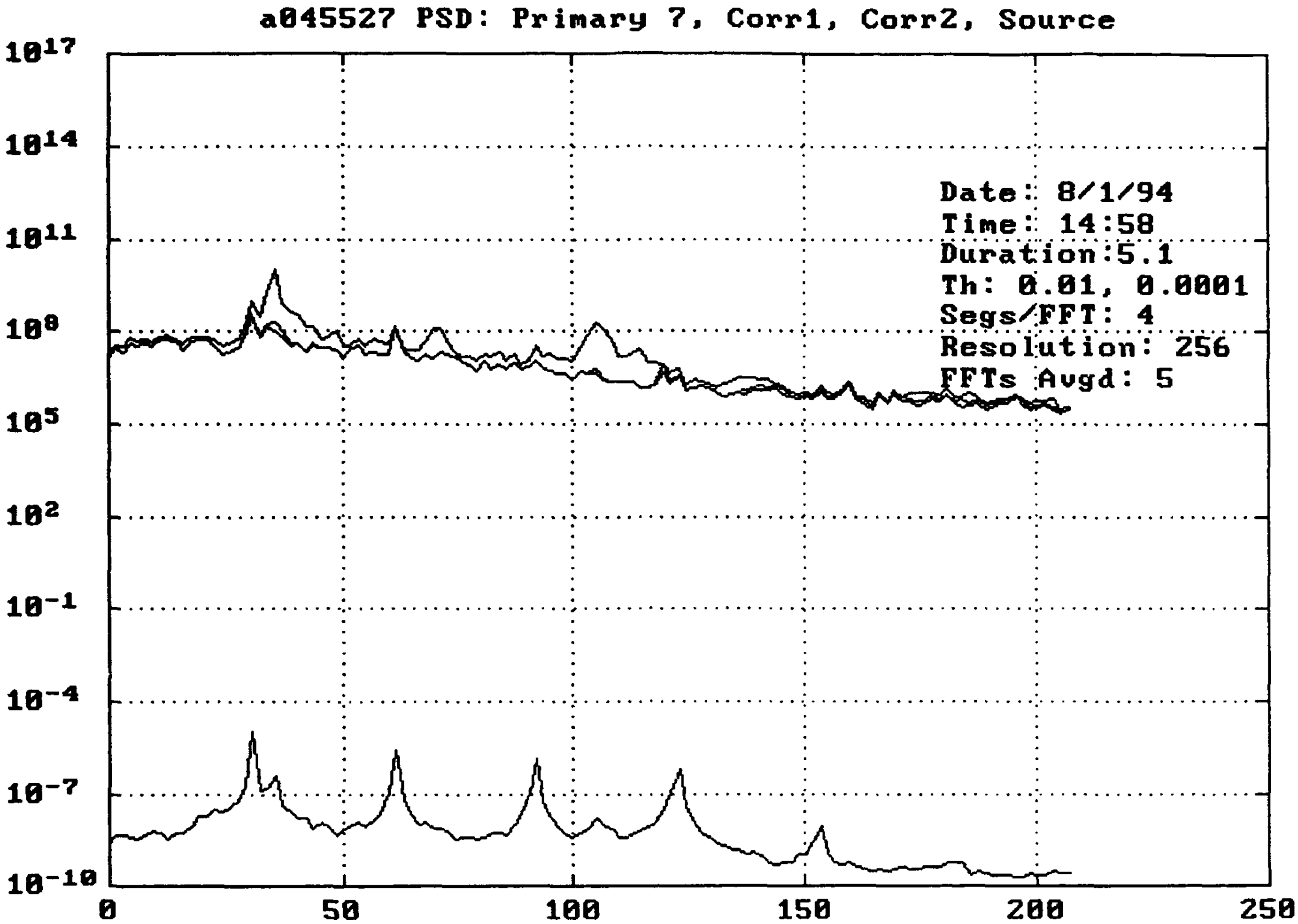


Figure VB-11. PSD's, Primary 7, corrected (2), source: 25 Hz sine wave

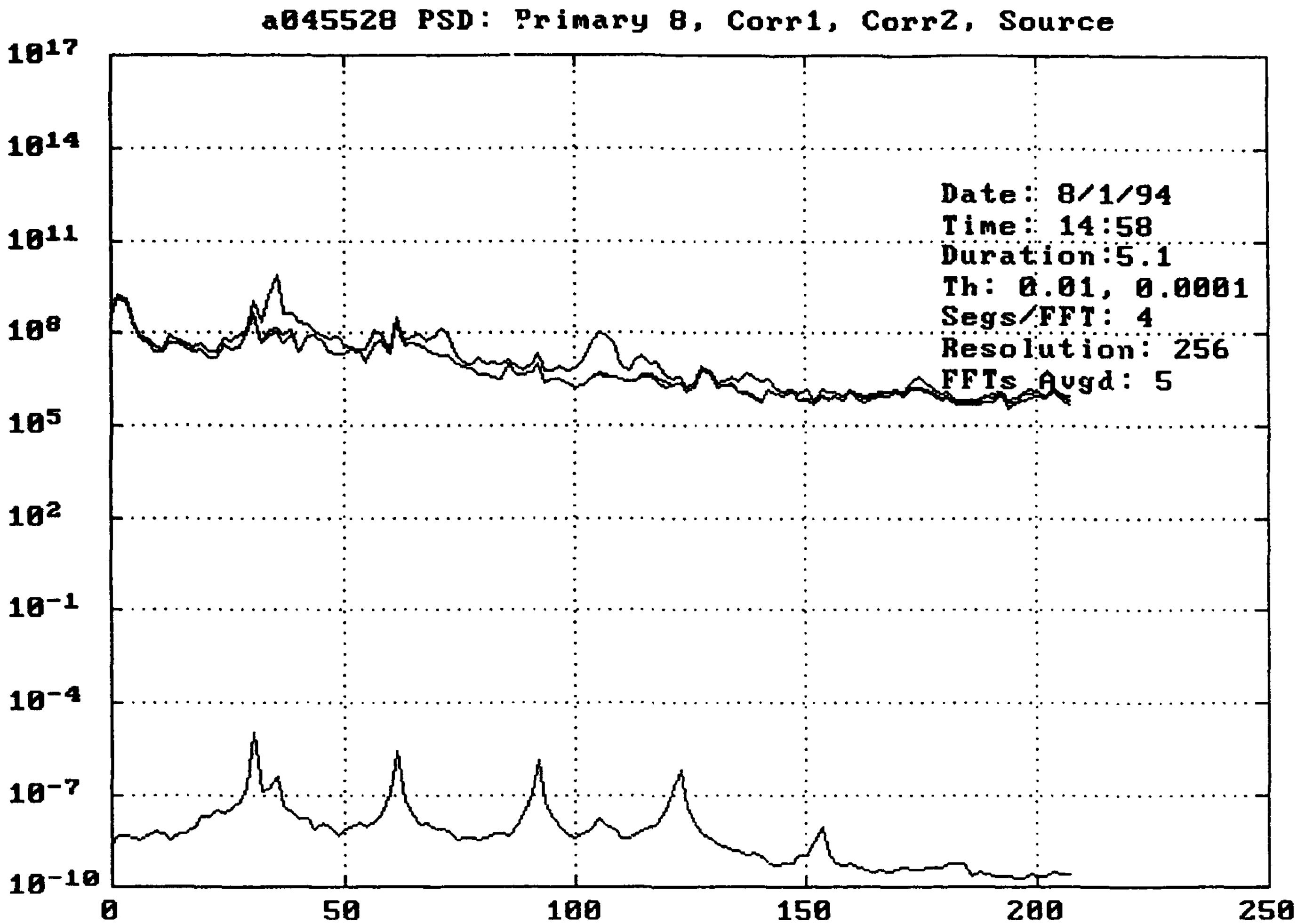


Figure VB-12. PSD's, Primary 8, corrected (2), source: 25 Hz sine wave

show substantial peaks at the several source frequencies, while neither the tailpipe microphone nor the three accelerometers exhibit this phenomenon.

Finally – and importantly – the coherence function of each reference with the source (Fig. VB-4) shows that the offending signal peaks in the reference power spectral densities in fact represented energy coherent with the source ("Thru" channel). Is coherence "transitive?" It is, somewhat, as indicated in Figure VB-5 showing the reference coherences with primary channel 3 instead. The same peaks do appear in these coherences as do in those taken with respect to the source itself.

Thus, the indicated solution in this experimental example was to omit the offending reference microphones and to only use the tailpipe microphone and the three accelerometers. The results for Primary 1 has already been shown in Figures V-1 and V-2; results for the remaining primaries (2-8) are shown in Figures VB-6 through 12, in overlaid fashion for easy visual interpretation of engine noise actually cancelled. Results for all four primary microphones facing the loudspeaker are very similar; the four on the far side show that the signal did not reach them very well. Primary microphones should, if possible, be mounted standing up from the roof rather than on the side of the vehicle, in order to provide more nearly omnidirectional sensitivity.

C. Simulated Vehicle Drive Bys - 1

A number of vehicle driveby recordings were analyzed. Little effective noise cancellation with signal retention was produced in any runs that we examined, because of the relatively large amount of signal source energy entering each reference. There were no good, i.e, sufficiently "signal-free," references eligible to provide noise cancellation, as was the case in the above illustrated test case and the sine wave recording.

To solve this problem, a more fully instrumented test should be held, in order to optimize the types and locations of the sensors, and most especially of the reference sensors. In such a test, the spectrum of all sensor responses to the source, as well as to the engine, should be measured. In addition, the coherence between the source and any sensor, and the coherence between pairs of sensors, should be measurable on the spot. Reference sensors should pick up the largest obtainable levels of engine noise so that their gains can be set low enough to exclude most of the signal from the loudspeakeres. All reasonable reference sensor location should be tested: in or on the exhaust system but

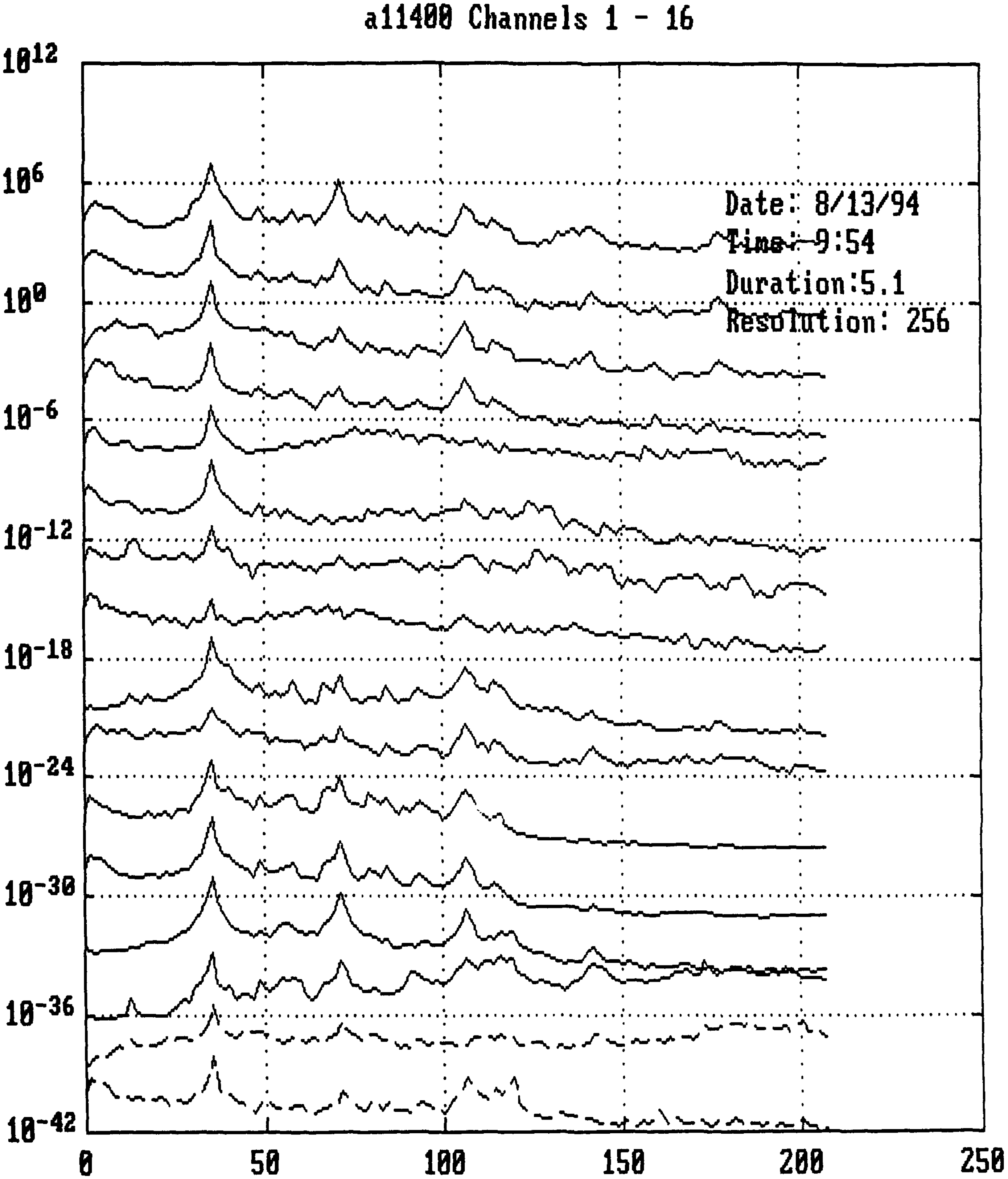


Figure VC-1. PSD's of all 16 channels: M60 tank, 15 km/hr

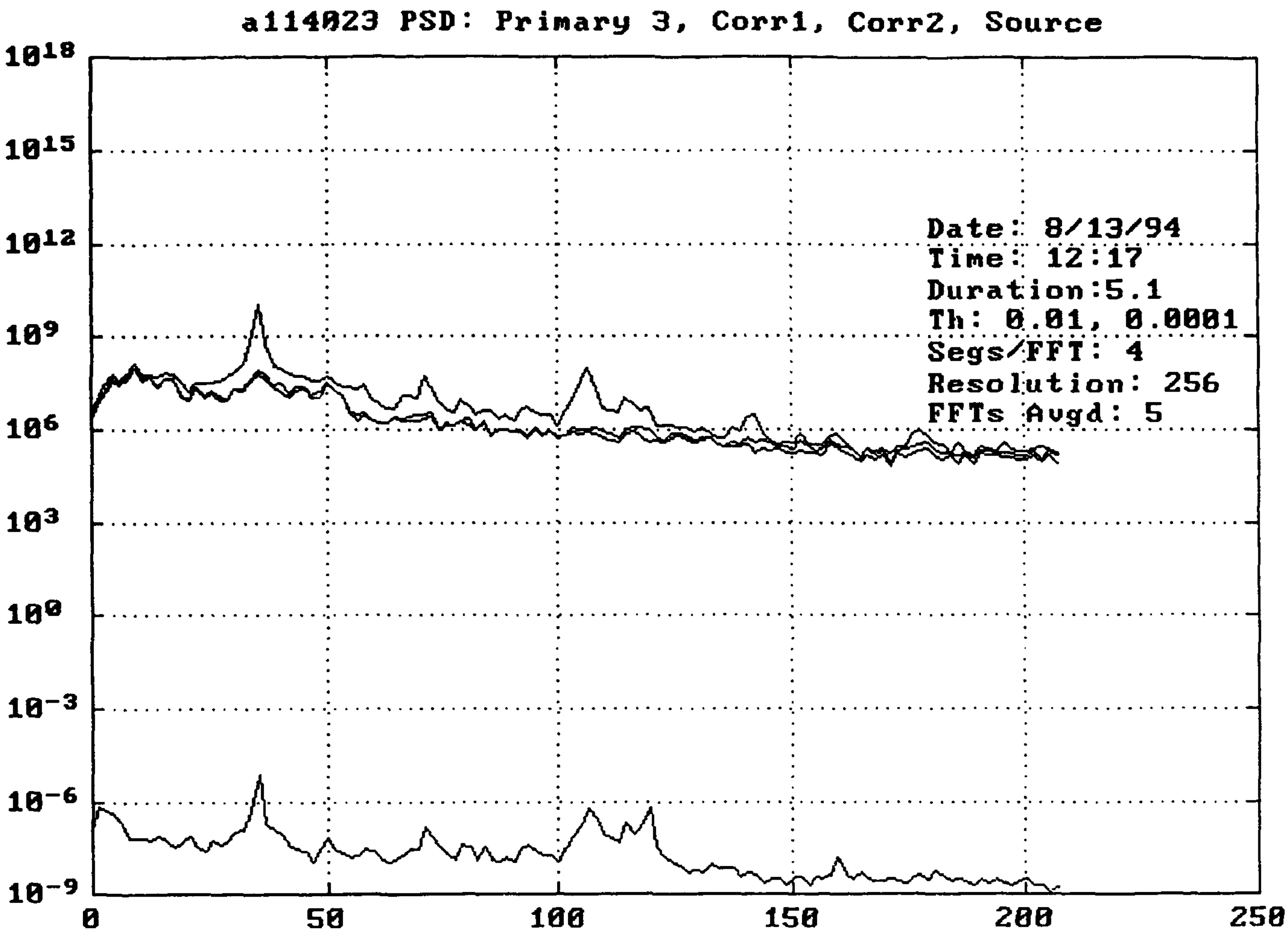


Figure VC-2. PSD's, Primary 3, corrected (2), source: M60 tank, 15 km/hr. All 7 references applied. Signal cancelled along with the noise, due to excessive signal in references.

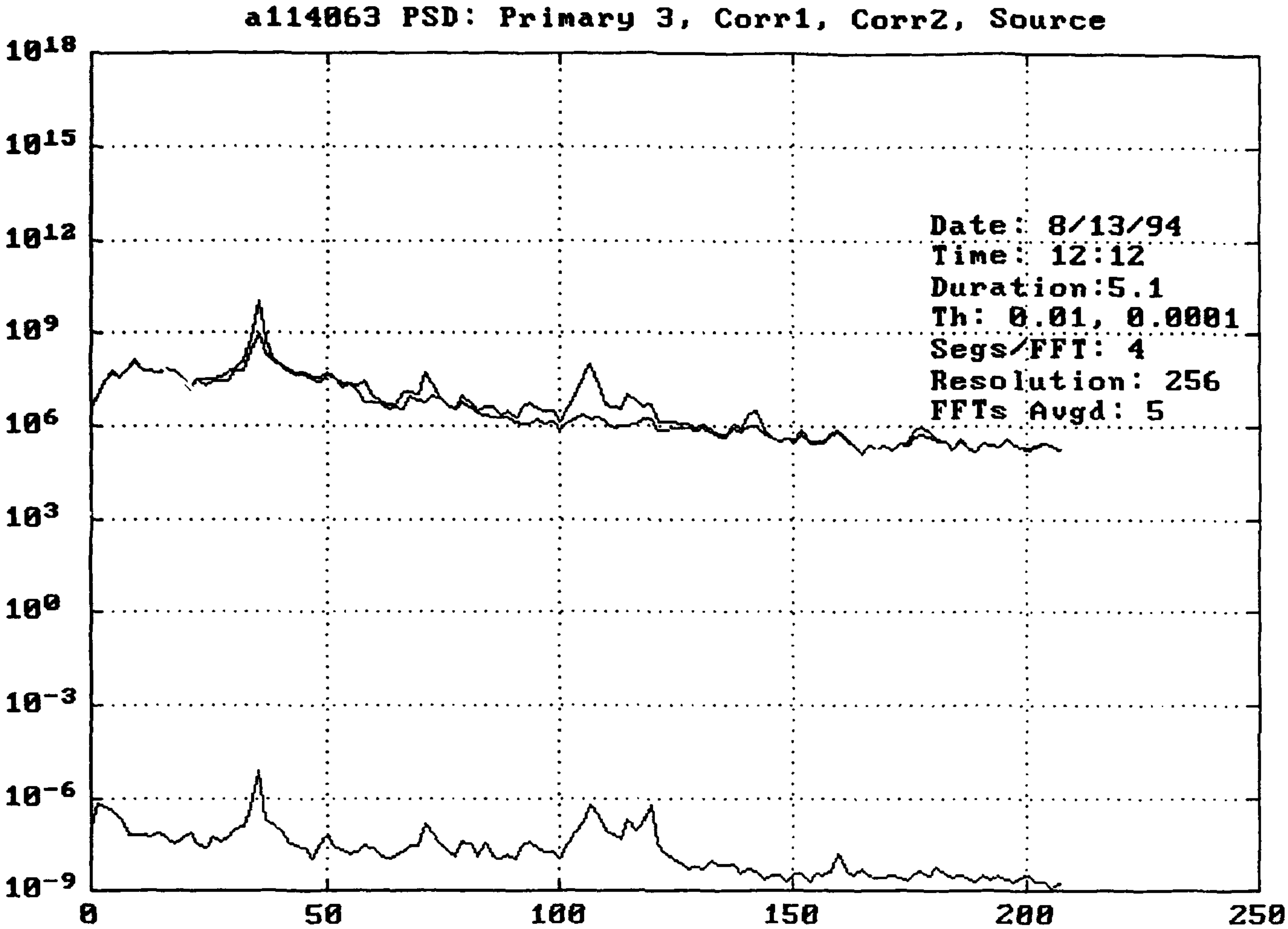


Figure VC-3. PSD's, Primary 3, corrected (2), source: M60 tank, 15 km/hr. Tailpipe reference only applied. Most of Signal cancelled along with the noise, due to excessive signal in references.

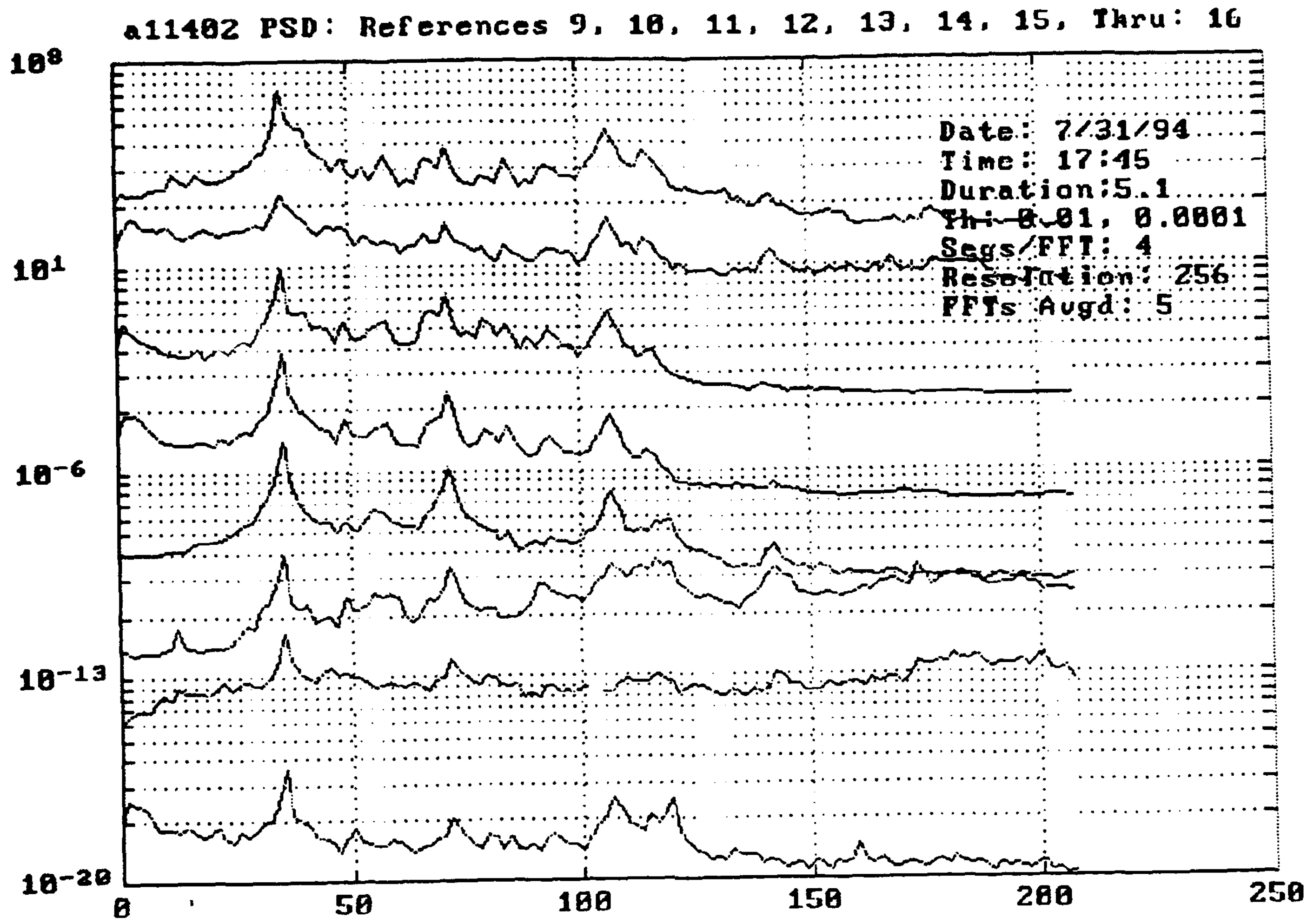


Figure VC-4. PSD's, References and source: M60 tank, 15 km/hr

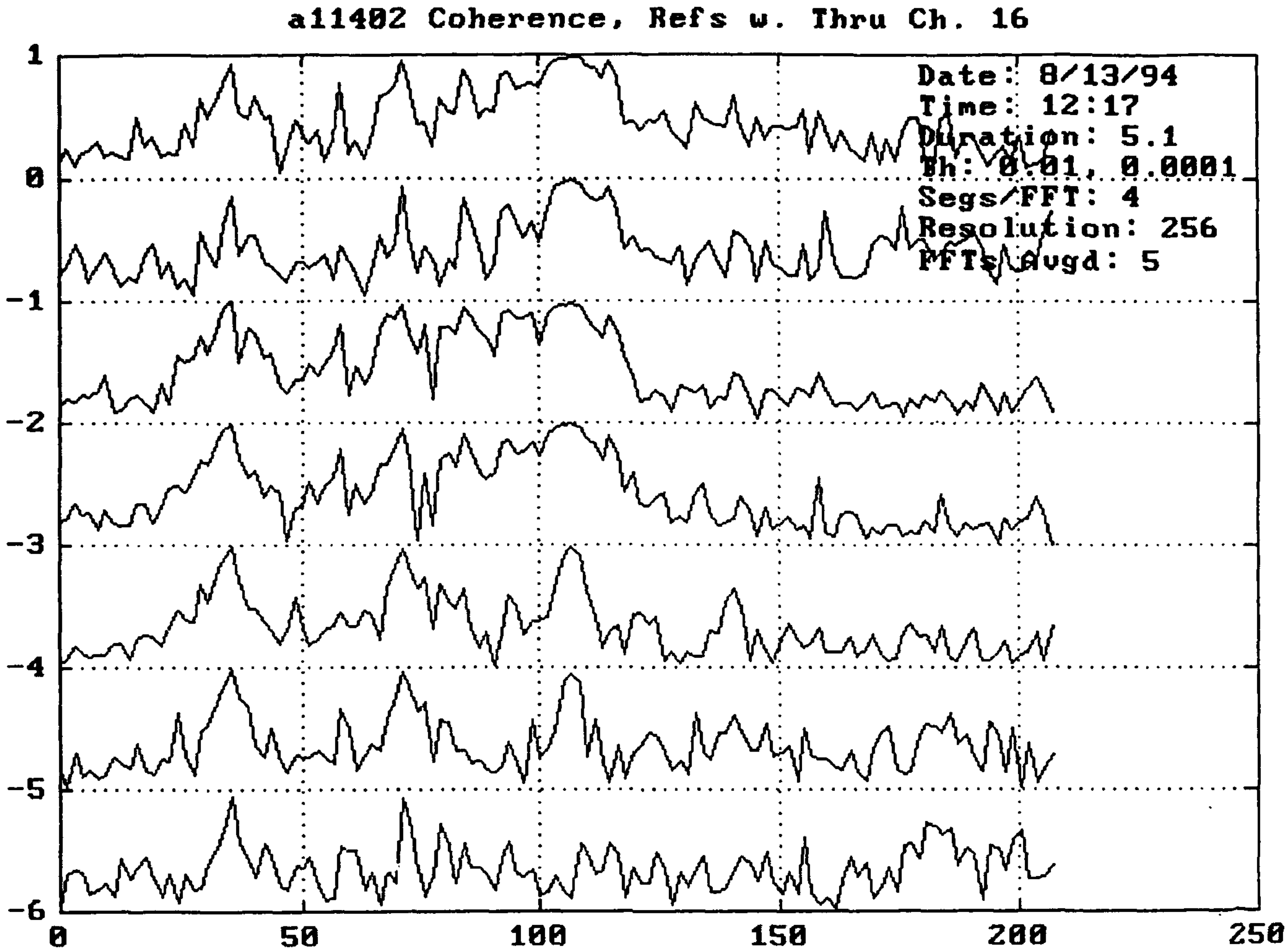


Figure VC-5. Coherences of references with source: M60 tank, 15 km/hr closer to the engine, at locations on the engine block or on the engine head, to mention just a few. These may well be accelerometers rather than microphones. The present location of the tailpipe microphone happens to have been on the side of the vehicle near the tailpipe and away from the loudspeakers; if the loudspeaker source were instead on the same side of the vehicle as the tailpipe, then the tailpipe microphone would probably receive even more loudspeaker signal than it does now. If omnidirectional response is desired for beamforming, then the primary microphones should stand up from the roof. There are lots of ideas – but they need to be tested.

Moreover, since nearfield source effects are unpredictable, the speakers should be placed sufficiently far away so that the vehicle will be in their far field. For example at 20 Hz, a wave length in air would be close to 60 feet away. All source levels and signal as well as noise levels received by the sensors should be carefully monitored and logged, and the noise sensor locations should be adjusted for best likely performance.

Files a11400, a11402, and a11406 playing the driveby of an M60 tank travelling at 15 km/hr present as good an example as any other. Figure VC-1 shows the power spectral densitities of all 16 channels (8 primary, 7 reference, and the source) for a 5.1 second record 19 minutes (1140 seconds) into Tape 1 or a. Figure VC-2 depicts in overlay fashion an attempt to cancel the noise in a typical primary sensor, in this case number 3, using all 7 references. Clearly lots of noise gets cancelled, but so does most of the signal. Figure VC-3 shows the slightly better but still very poor result when only the tailpipe microphone is utilized as a reference. Here, of the signal appearing in the uncorrected version at about 35 Hz, some 12 dB of it is thrown out by the cancelling process. A bit better than using all references, but not much. On the other hand, the all-references case does cancel considerably more noise than just the tailpipe microphone alone. Why and what to do?

Again the answer to the question lies in careful examination of the reference psd's and coherences with the source. Such examination reveals that all references have strong and coherent energy at the source signal peaks. It is a clear case of too much signal in the references – all of them, and their correct positions must be determined experimentally.

D. Simulated Vehicle Drive Bys - 2

It has been noted that Tape 5 was done with the HMMWV's engine turned off, and was therefore of somewhat lesser interest in connection with the present study. However, we wish to cite an example from it. We do this in part because this case at first caused us great concern about the correctness our software, until we found out what was happening. More importantly, this example dramatically illustrates the value in noise cancelling of examining coherences as well as power spectral densities of sensors.

Figure VD-1 depicts all 16 channels for Run e11400, some 1140 seconds into Tape 5 (or e). Again, we meet our old friend the M60 tank travelling 15 km/hr, but this time with the HMMWV's own engine turned off. Our concern in this run was with the prominent peak at 80 Hz. Since this peak does not appear on the last curve, the source spectrum, its real origin is not known with certainty because the recorded version of the source contains almost entirely what appears to be 60 Hz hum and its harmonics at 120 and 180 Hz. Thus, the recorded version of channel 16 did not correspond to what the loudspeakers appeared to have actually put out.

In any case, the attempted noise cancellation shown in Fig. VD-2 exhibits a blatant and stubborn refusal to cancel that 80 Hz peak, even though it shows up in force in the first four references, the reference microphones. As advertised, the coherence picture solves the puzzle. Figures VD-3 and 4 prove that whatever acoustic energy appears in the reference sensors at 80 Hz has a coherence of no more than 0.5 with either the recorded source (Fig. VD-3) or with Primary sensor 1 microphone itself (Fig. VD-4).

In this connection, it may be useful to review just what any given coherence ought to do for noise (or signal) cancellation. While this problem is complex for a multi-channel system, the relationship is straightforward for the special case of a single reference: The gain achievable against noise by such a reference is simply

$$\text{Gain} = -10 \log_{10} (1-r^2),$$

where r is the primary-reference coherence (square root form as plotted in our graphs). Thus, a single reference having a coherence of 0.5 with a primary channel could not have removed more than 1.25 dB of the signal or noise in question.

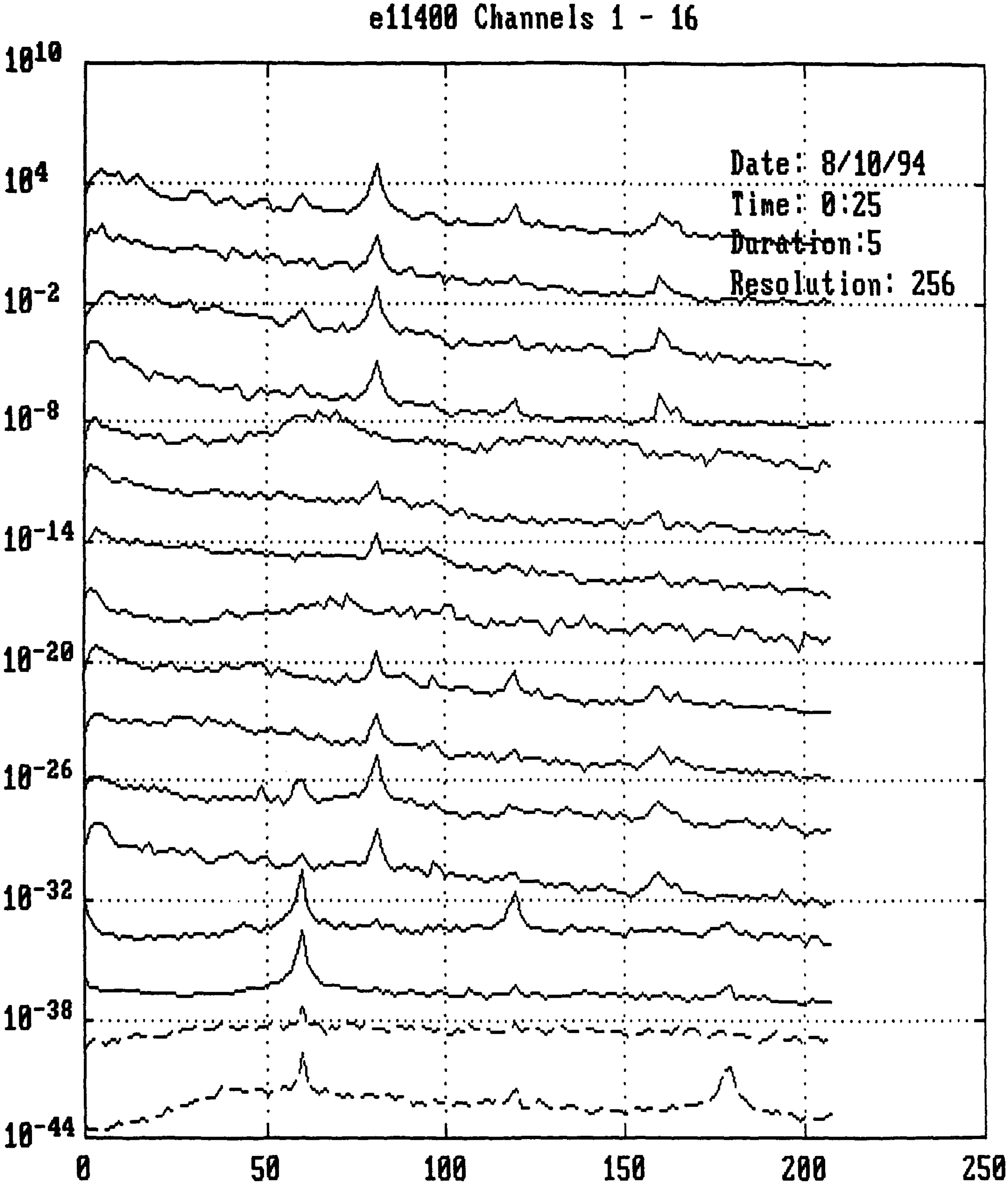


Figure VD-1. 16 channels: M60 tank, 15km/hr, own engine off. Note peak at 80 Hz in first 4 primaries and in first 4 references

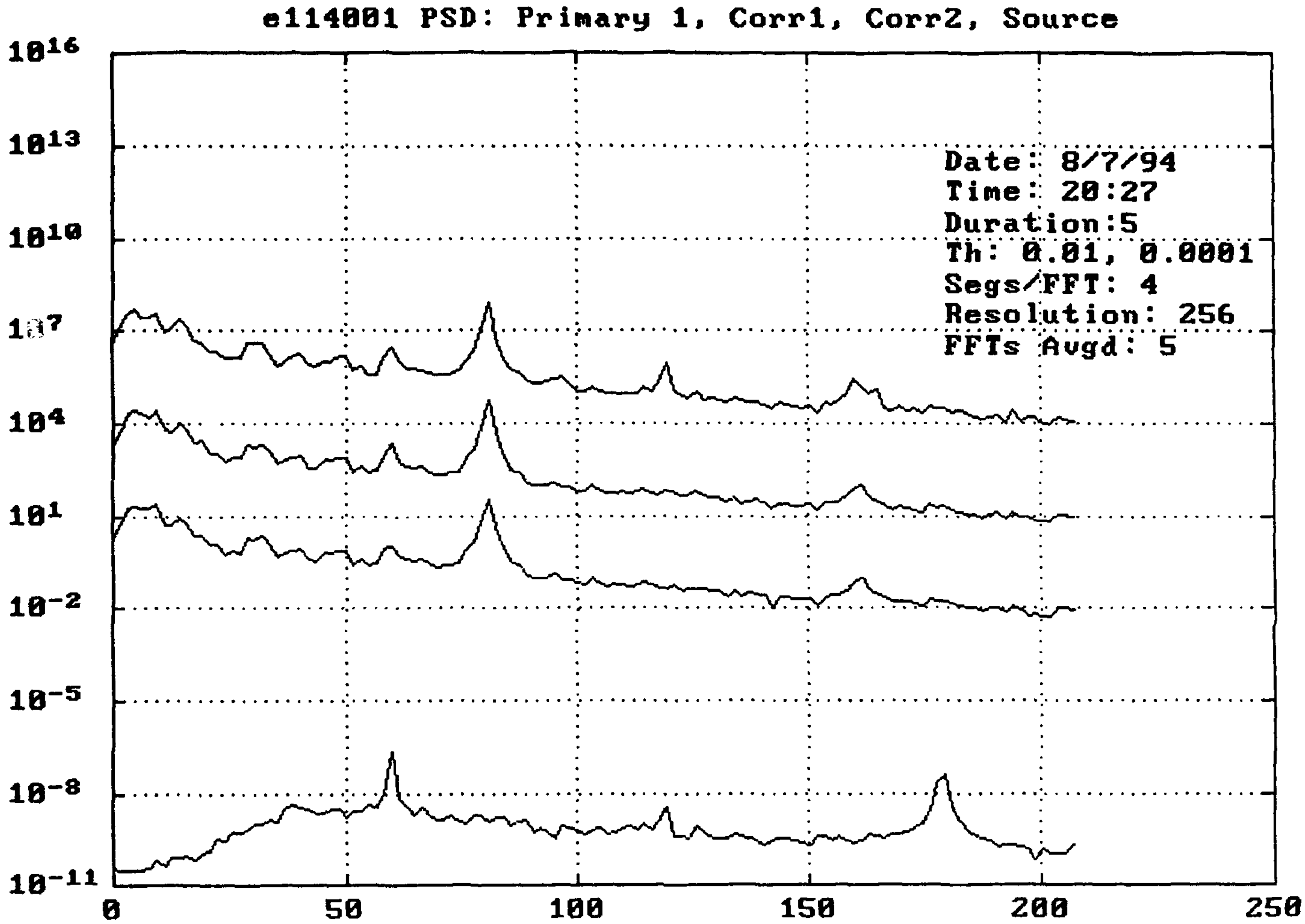


Figure VD-2. Failure to cancel peak at 80 Hz: M60 tank, 15km/hr, own engine off

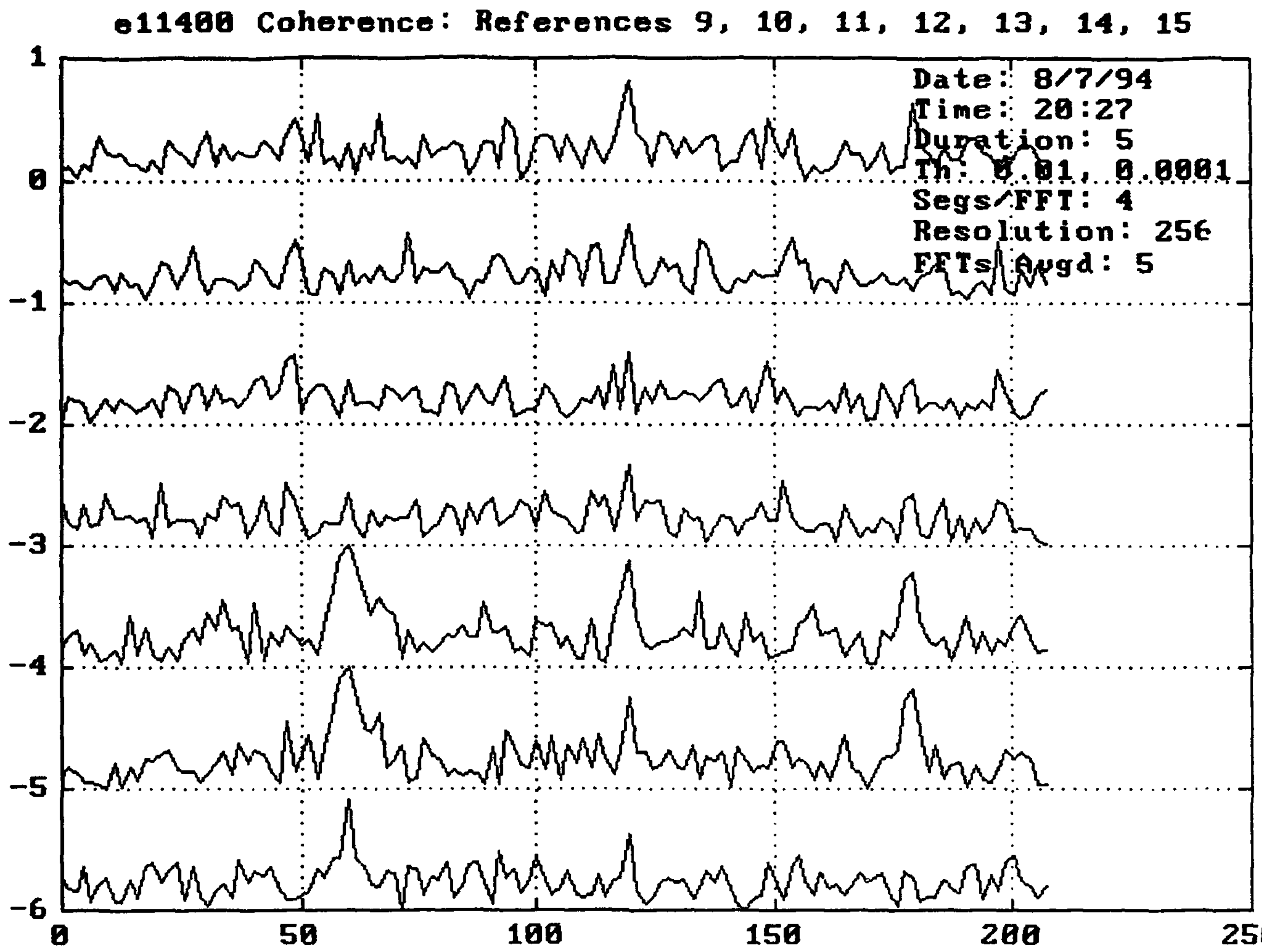


Figure VD-3. Coherence of all 7 references with the source.

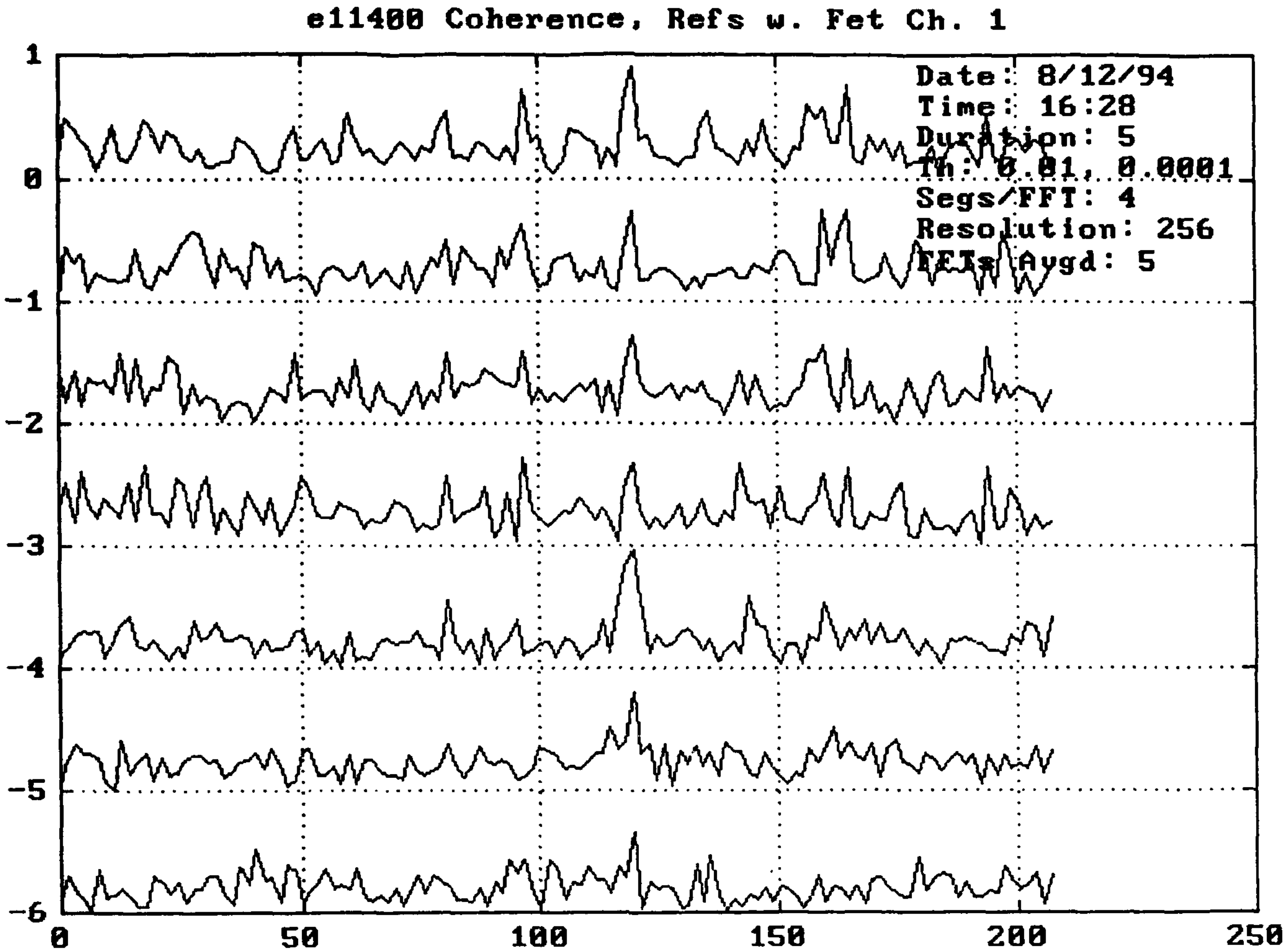


Figure VD-4. Lack of coherence at 80 Hz of all 7 references with Primary Microphone 1 explains failure in Fig. VD-2 to cancel peak at 80 Hz

VI. Conclusions

1. The experiments and analysis that have been performed have shown that the engine noise getting into the signal microphones of the test vehicle is readily cancelled.

2. The signal was received well on the side of the vehicle facing the loudspeakers, but more faintly on the far side. If omnidirectional sensor reception is desired for effective beamforming, the primary microphones should probably stand off the roof rather than be mounted on the side of the vehicle.

3. The signal has been preserved in some cases, but in most cases, its strong and coherent entrance into all references prevented signal retention as the noise was cancelled. Because of the particular coherences involved in the process, signal retention happened to be quite good for the 31 Hz tone, but poor for all the drivebys which were studied.

4. The noise sensor power spectral density (PSD) plots of the noise or reference sensors show that all these references have significant peaks of sound energy at precisely the frequencies at which the loudspeaker source had major peaks. This strongly suggested that significant loudspeaker signal was getting into each reference sensor.

5. The coherences plotted between each reference sensor and the source itself confirms the conclusion stated in 4: The PSD peaks appearing in the noise sensors at source frequencies do represent energy that is in fact coherent with the source energy at those frequencies, and should therefore be expected to be cancelled along with the noise.

6. When the data was taken, the greatest concern was to be sure to get the signal into the signal microphones. This has turned out not to be a problem at all; the problem to be solved is to not get too much of it into the noise sensors.

7. Judicious choice and location of noise sensors should remedy the above stated difficulty. Accelerometers would in general pick up less source signal than microphones. Locations to be considered would include further up the exhaust system, near the exhaust manifold, at various sites on the engine block and head, to mention a few. A tachometer on the engine, and a capability to produce the harmonics of the tachometer as possible reference inputs, could be useful. This technique has been used extensively in the active

cancellation of engine noise in aircraft cabins. A new test should include the capability to measure spectra as well as coherences for all sensors and sensor types, and sensor positions should be selectable during the tests.

8. In a new test series, more attention should be paid to getting a measure of absolute levels of signals and noise received by the various sensors, especially the noise sensors, and independently from either the engine or the source loudspeakers. The placement of the noise sensors should be adjusted experimentally in order to minimize the signal at their outputs. Also, in a new test, data file lengths should be kept a little more manageable: In the tests which have been concluded, each tape contained a single file pair which, when loaded into an IBM compatible computer, uses up the bulk of its entire disk space. The engine should be tested at a variety of speeds in addition to idling.

9. Assuming that appropriate noise sensor types and locations can be found to remove the above stated difficulties from the signal and noise measurements, the development of a real-time online noise canceller based on these principles would be well within existing technology, built primarily form commercial off-the-shelf (COTS) components.

VII. Recommendations

1. Perform another test series on the HMMWV vehicle, with appropriate spectrum and 2-channel coherence measurement capabilities, and with a wide selection of candidate reference sensor types and locations. The main purpose of such a new test will be to optimize the sensor configuration, especially the reference sensors. The test should be planned for at least two days rather than only one, in order to allow some off-line analysis of results before completing the series. The specific recommendations for such a new tests are listed in the following separate Section VIII.

2. Complete work on an adaptive code version making use of the Lockheed-Sanders data formats. Make the single-pass and adaptive codes more efficient and determine requirements for a digital signal processing board and other computer components to develop a real time multi-channel noise canceller.

3. Develop a real-time multi-channel noise canceller. Such a device would be useful also in other Army applications. Possible examples are noise-limited hand held endfire acoustic detection arrays under development at the Army Research Laboratories in Adelphi, MD, and an acoustic sensor project underway at the Army's Night Vision Electronic Sensors Directorate at Ft. Belvoir. Additionally, the development of a realtime multi-channel noise canceller would present a significant opportunity for technology transfer to commercial applications. Medical instrumentation, quieting aircraft cabins, and applications in the communications industry are likely candidates to welcome such a development.

VIII. Specific Recommendations for New Test Series

The new tests recommended as item 1 in Section VII above should be planned as follows:

1. The main purpose of new tests is to determine the best sensor types and locations, especially for the reference sensors.

2. Sensor types and positions should be changeable during the test.

3. Instrumentation should be available for the measurement of power spectral densities of all sensors, as well as the coherence between any two chosen sensors.

4. Reception of loudspeaker signal and engine self noise by all sensors should be monitored and relevant coherences examined, especially coherences between the references and the source.

5. Nonlinearities in the system should be minimized. In particular, the source should be placed far enough away to put the vehicle in its far field.

6. The primary sensors should stand off from the roof, rather than be mounted on the side of the vehicle.

7. Candidate reference sensor positions should include, but not be limited to, further up the tailpipe, mounted at various locations on the engine block and head, near the exhaust manifold, etc. The object is to get maximum noise level so the gain can be turned down enough to reject most of the signal.

8. More accelerometers should be tried, rather than reference microphones.

9. Any tailpipe sensor microphone should not be mounted on the side of the vehicle. It received signal there during the tests, and would have received more of it had the source been located on the same side of the vehicle as its tailpipe.

10. Primary sensor microphones should be mounted on vibration isolators to minimize noise getting into them.

11. Reference sensors should be isolated as much as possible from signal, not only through choice and location of the sensors but also through the judicious use of passive acoustic damping materials.

12. The engine should be operated at a larger selection of speeds, not only at idling.

13. The tests should be planned for at least two days rather than only one, to permit some offline analysis of the data before completion of the series.

Appendix A: Codes Used in the Calculations

1. System for naming data files

This section is presented primarily to facilitate reading the filenames printed on the title line in each plot. File types are identified by the file extensions .g1, .g2, .ecg, .fcg. of .psd as defined below. All plots are prepared from groups of ".psd" files.

1. .g1, .g2 files: These files are the original data files generated by Lockheed Sanders, Inc. test equipment and named by Lockheed Sanders personnel. These files are read by readlock.for to prepare an .ecg file for subsequent analysis. The files come in pairs .g1 and .g2, with .g1 containing eight channels of primary sensor data, and .g2 containing 7 channels of noise reference data and one through channel recording the loudspeaker source input. The sampling rate has been set at 2083.3 Hz. These files are long, about 45 Mbytes each, making them a bit unwieldy for use on a PC.

2. .ecg files: This is the original data, low-pass filtered and decimated by a factor of approximately 5 to yield 16 channels of data with a Nyquist cutoff frequency of 1/10 the sampling rate or 208.3 Hz. A synthetic .ecg file can be generated by program testfl1.for. A typical .ecg filename is 'a0455200.ecg.' This name identifies the run generating it as follows: The first digit, 'a', refers to the first of 5 data tapes. The next four digits '0455' denote the time in seconds from the beginning of the tape, according to the test log shown in Section V. Thus, '0455' means 455 seconds or 7 minutes and 35 seconds into Tape a, at about the middle of a 25 Hz sine wave recording. The sixth digit is also assigned to the .ecg file, but may be modified when the parameters used in the noise cancellation calculation are varied from their default values. A fifth digit other than "0" in the .ecg filename is used when readlock produces an .ecg file using other than the "standard" 5.1 second interval. A When .psd files are formed, the last two digits of the .ecg filename are overwritten as described below.

3. .fcg files: Normally used in electrocardiography and other applications where time domain information is required. Not used in this report.

4. .psd files: Power spectral density of reference sensors, source signal, primary sensors, and corrected primary sensors; coherence between each reference sensor and the source signal.

The last two digits of the file name pertain to the particular information in the file. If the e initial digits of the .ecg filename are followed by an 'i,' the file is an information file used to write a title and other information on the plot. A single numeric digit (7th of a 7 digit filename) refers to the number of a primary sensor from 1 to 8. A single digit followed by 'r' as the eighth digit refers to one of seven reference sensors; '1t' refers to the through or signal channel; while a single (reference sensor) digit followed by a 'c' refers to the coherence between the reference sensor and the signal input. Channels containing corrected primary data have as their 7th digit the number of the corrected primary channel, while the 8th digit denotes which of two thresholds was applied to the eigenvalues. The generally used default value of these thresholds were 20 and 40 dB respectively below the largest noise eigenvalue.

2. testfl1.for: Generation of synthetic data file to test calculation programs.

This program was written to verify the correctness of the noise cancellation calculations performed by armsp.for. Each of 7 simulated reference sensor signals (multiplexed channels 9 through 15) consisted of a single sine wave, each of a different frequency, whereas the 8 primary sensors were represented by various linear combinations of these reference signals. The "through" or "source" signal was taken as still another sine wave. The output format is identical to that of an .ecg file produced by readlock.for for real data.

3. readlock.for: Reading, low-pass filtering, and decimation of data files generated by Lockheed Sanders, Inc. test equipment

This program reads the two original data files (.g1 and g2), and performs the following functions on the data:

- The data is normalized to assure that each reference channel is given approximately the same weight in the calculations;

- A 128 point convolution filter is applied to filter out all except the lowest 20% of the frequencies contained in the original data, yielding a Nyquist cutoff frequency of 20% of half the 2083.3 sampling rate or 208.3 Hz.

- The data is decimated by a factor of 5, providing a new "sampling rate" of 416.7 Hz.

▸ The data is recorded in multiplexed fashion in the resulting .ecg file.

4. psd5.m, titl2.m: Plot power spectral densities and coherence

This code uses information from a collection of power-spectral density and related .psd files to generate the plots presented in this report. The numbers in the filenames pertain to the most recent version of the program. These programs require the use of Matlab for execution.

The code for performing the actual noise cancellation consists of code developed using private funds for the study of fetal electrocardiography, modified to accept the Lockheed Sanders, Inc. data format. This code is submitted to the Government as a proprietary addendum.

```
c readlock.for rev jcs: read as direct files
c readlock.for rev jcs; 940710 fr - formatting; fix write(31,); sampFreq
c readlock.for rev 940415 fr: rewind, nStatPoints
c readlock.for 940402 cs made points correspond to old calculated vals.
c readlock.for 940331 csfr (fix write to rec=6, reduced screen print)
c
c This is a simple program to read the data files from Lockheed Sanders.
c The data is in two byte signed integer format.  Output is a standard
c .ecg file, but with 16 channels.
c
c J. Clarke Stevens
c Signal Separation Technologies
c Feb 22, 1994.  Long Live George Washington!
c
c Main program block:
c
      program ReadLock
c
c Set constants:
c
      parameter (frequency = 2083.333333)   ! undecmtd sample frequency
      parameter (nChannels  = 16)         ! number of channels, 2 infiles
c      parameter (nChannels  = 8)          ! number of channels, 2 infiles
      parameter (nChannelsPerFile  = 8)   ! number of channels per file
      parameter (numWeights = 129) ! number of filter weights
c      parameter (nPoints     = 250880)     ! number of points per channel
c      parameter (nPoints     = 2000)     ! number of points per channel
c      parameter (nDataStart = 100)         ! position for data start
c      parameter (nStatStart = 200)
c      parameter (nDataStart = 3000)         ! position for data start
c      parameter (nStatStart = 3000)
c      parameter (nStatPoints = nPoints - nStatStart) ! points for stats
      parameter (iDecimator = 5)          ! decimation factor
c
      character*2  versfnm
      character*6  genfnm
      character*12 lokfname(2), ecgfname ! 2 input files and 1 .ecg file
      integer*2    idataPoint(nChannels,numWeights) ! array stores data
      integer*2    iData       ! point for calc of norm
      integer*4    i                ! loop index counter
      integer*4    iPoint                ! current point
      integer*4    iPosition             ! position for point in array
      integer*4    iChannel              ! current channel
      integer*4    iWeight          ! current filter weight
      integer*2    iNormalChannel   ! normalization reference chan
      integer*4    nStatPoints           ! number of points for stats
      integer*4    nPoints               ! length of interval in points
      integer*4    nDataStart            ! start position of data
      integer*4    nStatStart            ! start position of stats
      real*4       channelMean(nChannels)    ! mean vector
      real*4      channelRMS(nChannels)     ! RMS vector
      real*4      channelFactor(nChannels) ! normalization vector
      real*4      channelData(nChannels)    ! data from file
      real*4      referenceFactor   ! normalization reference
      real*4       filter(numWeights)     ! the filter
      real*4       filteredValue          ! value of new filtered point
      real*4       outputArray(nChannels) ! filtered data
c
c Initialize the filter:
c
      DATA filter /    ! fir2(128, 0.2)
     * -0.00006034715814,  -0.00009941091517,  -0.00012253653140,
     * -0.00012028676864,  -0.00009242607391,  -0.00004830372057,
     * -0.00000372571763,   0.00002516793517,    0.00002896915489,
     *    0.00000982576943,  -0.00001893206551,  -0.00003926645345,
```

```
     * -0.00003863192636,  -0.00001866189178,     0.00000340052880,
     *    0.00000334184665,  -0.00003407792675,  -0.00010018672209,
     * -0.00015880652690,  -0.00016051815368,  -0.00007217562514,
     *    0.00009142335120,   0.00025274476980,   0.00028910272145,
     *    0.00008429697520,  -0.00040324181965,  -0.00108862727079,
     * -0.00175946196572,  -0.00214313945983,  -0.00202871289202,
     * -0.00139646475704,  -0.00049056576074,     0.00021892251965,
     *    0.00019721237613,  -0.00091662052100,  -0.00308955168638,
     * -0.00580425027097,  -0.00817446923883,  -0.00926214962530,
     * -0.00849584334273,  -0.00601968361042,  -0.00279632750809,
     * -0.00036200983333,  -0.00026871329733,  -0.00339705574340,
     * -0.00941878856097,  -0.01667716091679,  -0.02262234188681,
     * -0.02472390264145,  -0.02156211412970,  -0.01366835750137,
     * -0.00371252253126,   0.00416483462586,     0.00576261007593,
     * -0.00145935733668,  -0.01697073075788,  -0.03666996979762,
     * -0.05362766771134,  -0.05998681105141,  -0.04949498879655,
     * -0.01985468869073,   0.02594069073519,     0.07983730167155,
     *    0.13057277352257,   0.16665732094943,   0.17958793692696,
     *    0.16628817069530,   0.12999470587605,   0.07930730576657,
     *    0.02571109266601,  -0.01963496700252,  -0.04883733502557,
     * -0.05905609841986,  -0.05267563950488,  -0.03593652815811,
     * -0.01659290857436,  -0.00142354365837,     0.00560796483330,
     *    0.00404342501426,  -0.00359562105677,  -0.01320568141090,
     * -0.02078073134310,  -0.02376816501575,  -0.02169243184358,
     * -0.01595021289488,  -0.00898448963124,  -0.00323172193383,
     * -0.00025493587607,  -0.00034249065892,  -0.00263801670633,
     * -0.00566235833030,  -0.00796775203712,  -0.00865997387457,
     * -0.00761917668506,  -0.00539268295469,  -0.00286107650384,
     * -0.00084598415796,   0.00018138751069,     0.00020064334689,
     * -0.00044797321677,  -0.00127046810934,  -0.00183861716398,
     * -0.00193471072745,  -0.00158197279847,  -0.00097479230527,
     * -0.00035956320142,   0.00007484553011,     0.00025557837650,
     *    0.00022246161379,   0.00008011732734,  -0.00006297506050,
     * -0.00013945872498,  -0.00013740345923,  -0.00008634753125,
     * -0.00002926680768,   0.00000286131625,     0.00000290466756,
     * -0.00001591721593,  -0.00003294034542,  -0.00003352141111,
     * -0.00001621158932,   0.00000845869992,     0.00002513869422,
     *    0.00002208248157,  -0.00000331626737,  -0.00004376782194,
     * -0.00008553486229,  -0.00011402362221,  -0.00011923516930/

c       DATA filter /           ! fir1(32, 0.18)
c      *  0.00058458747668,  0.00152178621646,  0.00260408350366,
c      *  0.00337243944497,  0.00273793424247, -0.00051567165154,
c      * -0.00679459801153, -0.01477347432336, -0.02105847894341,
c      * -0.02082849374397, -0.00942595730214,  0.01561605309829,
c      *  0.05293659939349,  0.09689033004366,  0.13865664112702,
c      *  0.16867606142004,  0.17960031601842,  0.16867606142004,
c      *  0.13865664112702,  0.09689033004366,  0.05293659939349,
c      *  0.01561605309829, -0.00942595730214, -0.02082849374397,
c      * -0.02105847894341, -0.01477347432336, -0.00679459801153,
c      * -0.00051567165154,  0.00273793424247,  0.00337243944497,
c      *  0.00260408350366,  0.00152178621646,     0.00058458747668/
c
c       DATA filter  /          ! fir1(32, 0.20)
c      * -0.00093504638020,  0.0              ,  0.00153635353578,
c      *  0.00366652042354,  0.00541457094399,  0.00483568063737,
c      *  0.0              , -0.00935583412396, -0.02042464712700,
c      * -0.02722169079728, -0.02231751180261,  0.0              ,
c      *  0.04045333789719,  0.09304319617535,  0.14599607133856,
c      *  0.18535634387163,  0.19990531081529,  0.18535634387163,
c      *  0.14599607133856,  0.09304319617535,  0.04045333789719,
c      *  0.0              , -0.02231751180261, -0.02722169079728,
c      *-0.02042464712700,  -0.00935583412396,  0.0              ,
c      * 0.00483568063737,   0.00541457094399,  0.00366652042354,
c      * 0.00153635353578,   0.0              , -0.00093504638020/
```

```
c
c      DATA filter  /        ! fir2(?,?,?)
c     * -0.00118336176206, -0.00089131594153, -0.00001941687929,
c     *  0.00183307358786,  0.00447381421810,  0.00655863098588,
c     *  0.00578322229951,  0.00008251235030, -0.01058137040957,
c     * -0.02284564402156, -0.03006243555378, -0.02437272582921,
c     * -0.00016635241514,  0.04257092099101,  0.09676837531446,
c     *  0.14970980033216,  0.18713805956575,  0.19842583197463,
c     *  0.18058534273937,  0.13933525341297,  0.08676893723930,
c     *  0.03671403844390, -0.00013766127276, -0.01929266627777,
c     * -0.02267169800457, -0.01633439041741, -0.00713195380247,
c     *  0.00005212477956,  0.00341196076046,  0.00363222481224,
c     *  0.00239629063859,  0.00103379819657, -0.00001350558395/
c
c Enter in- and outfile names:
c
      write(*,51)
   51 format(' Enter first  input (lokfile) name: '$)
      read (*,'(a12)') lokfname(1)
      if (nChannels.gt.nChannelsPerFile) then
        write(*,52)
   52   format(' Enter second input (lokfile) name: '$)
        read (*,'(a12)') lokfname(2)
      endif
      write(*,53)
   53 format(' Enter Generic (no extension) .ecg file name, 6 char.: '$)
      read (*,'(a6)') genfnm
      write(*,54)
   54 format(' Enter Version name, 2 char. to append to generic:  '$)
      read (*,'(a2)') versfnm
      ecgfname = genfnm//versfnm//'.ecg'
      write(*,55)
   55 format(' Write filtered array on screen? (0=no, 1=yes) ? '$)
      read(*,*) iscreen
      write(*,56)
   56 format(' Enter norm. reference channel # (0=no norm):  '$)
      read(*,*) iNormalChannel
      write(*,57)
   57 format(' Enter start time in seconds:  '$)
      read(*,*) startTime
      nDataStart = frequency * startTime
      nStatStart = nDataStart
      write(*,58)
   58 format(' Enter interval length in seconds:  '$)
      read(*,*) timeInterval
      nPoints = frequency * timeInterval
      nStatPoints = nPoints
c
c Open 2 files for reading, 1 for writing:
c
      open(21, file=lokfname(1), status='old', form='binary',
     *         access='direct',recl=2)

      if (nChannels.gt.nChannelsPerFile) then
        open(22, file=lokfname(2), status='old', form='binary',
     *       access='direct',recl=2)
      endif

      open(31, file=ecgfname,status='new',access='direct',
     *         form='unformatted',recl=64)
c
c Provide header information
c
      dummy=0.0
c     write(31,rec=1) pname(1)
```

```
c     write(31,rec=2) pname(2)
c     write(31,rec=3) refphys(1)
c     write(31,rec=4) refphys(2)
c     write(31,rec=5) (ecgdate(i),ecgtime(i),i=1,2)
c     write(31,rec=6) (age(i),i=1,2),nchans,ndisk,timed,isampf,icombine
c     write(31,rec=7) (ysclchg(i),i=1,8)
c     write(31,rec=8) (ysclchg(i),i=9,16)
C     WRITING:     DUMMY,DUMMY,NCHNSE,NCHNSF, TIMED   ,   SAMPF    ,IDUM
      sampFreq = frequency/idecimator
      write(31,rec=6) 0.0,0.0,16,32,timeInterval,sampFreq,0
      write(31,rec=7) (1.0,i=1,8)
      write(31,rec=8) (1.0,i=9,16)
c
c Initialize, zero out the data array
c
      do 50 iChannel = 1, nChannels
      do 50 i = 1, numWeights
        idataPoint(iChannel, i) = 0
   50 continue
      irec = 11       ! initialize record number for .ecg file
      do 350 iChannel = 1, nChannels
        channelMean(iChannel) = 0.0
      channelRMS(iChannel) = 0.0
      channelFactor(iChannel) = 1.0
  350 continue

      if ((iNormalChannel.ge.1).AND.(iNormalChannel.le.nChannels)) then
c
c read points to skip to start position
c
c        do 360 iPoint = 1, nStatStart
c        do 360 iChannel = 1, nChannels
c          if (iChannel.le.nChannelsPerFile) then
c            read(21) iData
c          else
c            read(22) iData
c          endif
c  360   continue
        if (nStatStart.gt.0) then
          read(21,rec=nStatStart*nChannelsPerFile) iData
          read(22,rec=nStatStart*nChannelsPerFile) iData
        endif
c
c Calculate the normalization factors
c Use several points to compute the normalization factors
c
        do 400 iPoint = 1, nStatPoints
        do 400 iChannel = 1, nChannels
          if (iChannel.le.nChannelsPerFile) then
            read(21) iData
          else
            read(22) iData
          endif
          channelMean(iChannel) = channelMean(iChannel) + real(iData)
  400   continue
c
c Now calculate the mean values
c
        do 405 iChannel = 1, nChannels
          channelMean(iChannel) = channelMean(iChannel) /
     *                            real(nStatPoints)
  405   continue
        rewind(21)
        if (nChannels.gt.nChannelsPerFile) then
          rewind(22)
```

```
        endif
      endif
c
c Position input files correctly
c
c      do 460 iPoint = 1, nDataStart
c      do 460 iChannel = 1, nChannels
c        if (iChannel.le.nChannelsPerFile) then
c          read(21) iData
c        else
c          read(22) iData
c        endif
c  460 continue

      if (nDataStart.gt.0) then
        read(21,rec=nDataStart*nChannelsPerFile) iData
        read(22,rec=nDataStart*nChannelsPerFile) iData
      endif
c
c Read each point in sequence and convolve
c
      do 500 iPoint = 1, nPoints            ! convolve @ each point
        iPosition = mod(iPoint - 1, numWeights) + 1
c
c read the data points
c
        do 150 iChannel = 1, nChannels ! read each channel, 2 files
          if (iChannel.le.nChannelsPerFile) then
            read(21) idataPoint(iChannel, iPosition)
          else
            read(22) idataPoint(iChannel, iPosition)
          endif
  150   continue
c
c process the data if at decimated point
c
        if (mod(iPoint, iDecimator) .eq. 1) then
c
c convolve the filter and data stream
c
          do 300 iChannel = 1, nChannels
            filteredValue = 0.0        ! initialize
            do 250 iWeight = 1, numWeights
              filteredValue = filteredValue + filter(iWeight) *
     *          (real(idataPoint(iChannel, mod(iPosition - iWeight +
     *           numWeights, numWeights) + 1)) - channelMean(iChannel))
c    *           * channelFactor(iChannel)
  250       continue
            outputArray(iChannel) = filteredValue
            channelRMS(iChannel) = channelRMS(iChannel) +
     *        ((filteredValue**2) / real(nPoints / iDecimator))
  300     continue

cjcs calculation of mod changed to mod operator
c Only look at the values every thousand points.
c
          if ((iscreen.ne.0) .and.
     *      (mod(irec,(1000 / iDecimator) * iDecimator) .eq. 11)) then
            write(*,*) irec
c
c Print the data points; close files when done:
c
            write(*,61) iPoint, nPoints
   61       format(' Point ', i7, ' of ', i7)
            write(*, 60) (outputArray(i), i=1, nChannels)
```

```
            write(*,*)
   60       format(8(1x,e8.2))
          endif
          write(31, rec=irec) (outputArray(i), i=1, nChannels)
cfr          write(31, rec=irec) (outputArray(nChannels - i + 1),
cfr     *                           i=1, nChannels)
          irec = irec + 1                   ! next record
        endif
  500 continue
      close(21)                         ! close the input files
      if (nChannels.gt.nChannelsPerFile) then
        close(22)
      endif
      write(*,517) iNormalChannel, channelRMS(iNormalChannel)
  517 format(' Normal channel ',i2,' MS value: ',f12.2)
      referenceFactor = sqrt(channelRMS(iNormalChannel))
c
c Now calculate the normalization factors
c
      do 550 iChannel = 1, nChannels
        write(*,518) iChannel, channelRMS(iChannel)
  518   format(' Channel ',i2,' MS value: ',f12.2)
        channelRMS(iChannel) = sqrt(channelRMS(iChannel))
        if (channelRMS(iChannel).eq.0.0) then
          channelFactor(iChannel) = 1.0
        else
          channelFactor(iChannel) = referenceFactor /
     *                              channelRMS(iChannel)
        endif
        write(*,522) iChannel, channelRMS(iChannel)
  522   format(' Channel ',i2,' RMS value: ',f7.2)
        write(*,525) iChannel, channelFactor(iChannel)
  525   format(' Channel ',i2,' factor: ',f7.2)
  550 continue
c
c Write out scale factors
c
      write(31,rec=7) (channelFactor(i),i=1,8)
      write(31,rec=8) (channelFactor(i),i=9,16)
      write(*,*) ' Normalizing...'
c
c skip file header
c
      irec = 11
c multiply data by normalization factors
      do 600 iPoint = 1, nPoints / iDecimator
      read(31, rec=irec) (channelData(i),i=1,nChannels)
      do 620 iChannel = 1, nChannels
        channelData(iChannel) = channelData(iChannel) *
     *                          channelFactor(iChannel)
  620 continue
      write(31, rec=irec) (channelData(i),i=1,nChannels)
      irec = irec + 1                   ! next position
  600 continue
      close(31)                         ! close the output file
      end
```

```
% psd5.m rev 940731 frcs incl. coherence plots
%
% matlab program to make power spectral density plots from ascii
% .psd files.
% For any ECG for which .psd files have been prepared, there are:
%   nfets primary files 1 through <nfets>,
%   2*nfets corrected primary files, with a pair of files for each of
%   nfets primary sensors.
%   nthru throuput files 1t through <nthru>t
%

% get 6 char. Generic filename, Load info file
nam=input('Enter generic 6-char fcg filename in single quotes : ');
namei = [nam, 'I'];

eval(['load c:\army\armspcod\',namei,'.PSD']);
info = eval(namei);

% get the information from the info file
plotdate = [num2str(info(1)),'/',num2str(info(2)),'/',num2str(info(3))];
if (info(5) < 10);
  plottime = [num2str(info(4)),':0',num2str(info(5))];
else;
  plottime = [num2str(info(4)),':',num2str(info(5))];
end;
nfets = info(6);
nrefs = info(7);
nthru = info(8);
avgtime  = info(9);
thresh1  = info(10);
thresh2  = info(11);
FFTres   = info(12);
fpts = FFTres / 2 + 1;
segsPerFFT = info(13);
FFTAvgCount = info(14);

% generate frequency vector
for i=1:fpts;
  freq(i)=(i-1)*208.333333/fpts;
end;

moreplots = 1;
while (moreplots == 1);

  % What do you want to plot -- Refs or Primaries
  nrf=input(['Plot Ref(s) & Thru (1), or a Primary, Corrs, '...
     '& Thru (2), Coherence (3) ? ']);

  % Case of References & Thru
  if (nrf == 1),
    % Build thru file name; load file
                    for nt = 1:nthru;
      namet = [nam, int2str(nt),'T'];
      eval(['load c:\army\armspcod\',namet,'.PSD']);
    end;

    % Build ref file names, load files:
    for nr=1:nrefs,
      namer(nr,:) = [nam, int2str(nr),'R'];   % build the file name
         numr(nr) = info(14 + nr);            % getthe plot ID
      eval(['load c:\army\armspcod\',namer(nr,:),'.PSD']) % load file

          % get the vectors for each trace
      plotXVector(nr,:) = freq;
      temp = abs(eval(namer(nr,:)))/(10^(nr*3));
```

```
      plotYVector(nr,:) = temp';
    end;

                        % add the thru channel
    plotXVector(nrefs + 1,:) = freq;
    temp = abs(eval(namet))/(1e26);
    plotYVector(nrefs + 1,:) = temp';

         % draw the plot
    semilogy(plotXVector',plotYVector');
    titl2;
%    topline=0.7*10^6;
    topline=0.7*10^3; % for testfile
    separation=-1;
    text(180,topline*10^separation,['Date: ',plotdate]);
        text(180,topline*10^(separation-1),['Time: ',plottime]);
        text(180,topline*10^(separation-2),['Duration:',...
          num2str(avgtime)]);
        text(180,topline*10^(separation-3),['Th: ',...
          num2str(thresh1),', ',num2str(thresh2)]);
        text(180,topline*10^(separation-4),['Segs/FFT: ',...
          num2str(segsPerFFT)]);
        text(180,topline*10^(separation-5),['Resolution: ',...
          num2str(FFTres)]);
        text(180,topline*10^(separation-6),['FFTs Avgd: ',...
          num2str(FFTAvgCount)]);
    pause;
  elseif (nrf == 2),    % Case of Primaries, Correcteds & Thru

    morefets = 1;
    while (morefets == 1);
      % build other file names
      nf=input(['Enter primary channel: ']);
      s =input(['Enter plot separation (0=min, 4=max): ']);

      name1 = [nam, int2str(nf)];
      name2 = [nam, int2str(nf),'1'];
      name3 = [nam, int2str(nf),'2'];
      name4 = [nam, int2str(1),'T'];

      % Load Primary, Corrected, & Thru channels
      eval(['load c:\army\armspcod\',name1,'.PSD']);
      eval(['load c:\army\armspcod\',name2,'.PSD']);
      eval(['load c:\army\armspcod\',name3,'.PSD']);
      eval(['load c:\army\armspcod\',name4,'.PSD']);

      % plot the Pri., Corrected, and Thru PSD's
%%      semilogy(freq,abs(eval(name1)),freq,abs(eval(name2))/10^s,...
%%        freq,abs(eval(name3))/10^(2*s),freq,abs(eval(name4))/...
%%        10^(3*s+6));
      semilogy(freq,abs(eval(name1)),freq,abs(eval(name2)),...
        freq,abs(eval(name3)),freq,abs(eval(name4))/...
        10^(3*s+6));
      titl2;    % put a title and grid on the plot
%      topline=0.7*10^16;
      topline=10^13;  % for testfile
      separation=-1;
      text(180,topline*10^separation,['Date: ',plotdate]);
        text(180,topline*10^(separation-1),['Time: ',plottime]);
        text(180,topline*10^(separation-2),['Duration:',...
          num2str(avgtime)]);
        text(180,topline*10^(separation-3),['Th: ',...
          num2str(thresh1),', ',num2str(thresh2)]);
        text(180,topline*10^(separation-4),['Segs/FFT: ',...
          num2str(segsPerFFT)]);
```

```
        text(180,topline*10^(separation-5),['Resolution: ',...
          num2str(FFTres)]);
        text(180,topline*10^(separation-6),['FFTs Avgd: ',...
          num2str(FFTAvgCount)]);
      pause;    % wait for keypress
      morefets  = input(['Plot more Primary Plots (0=no, 1=yes) ? ']);
    end; % end of while morefets == 1

  % Case of coherence plots
  elseif (nrf == 3),
    % Build ref file names, load files:
    for nr=1:nrefs,
      namec(nr,:) = [nam, int2str(nr),'C'];   % build the file name
      numc(nr) = info(14 + nr);               % get the plot ID
      eval(['load c:\army\armspcod\',namec(nr,:),'.PSD']) % load file

      % get the vectors for each trace
      plotXVector(nr,:) = freq;
      temp = abs(eval(namec(nr,:))) - nr + 1;
      plotYVector(nr,:) = temp';
    end;

    % draw the plot
    plot(plotXVector',plotYVector');

    titl2;
    topline=0.7;
    separation=-0.25;
    text(180,topline,['Date: ',plotdate]);
    text(180,topline+(separation*1),['Time: ',plottime]);
    text(180,topline+(separation*2),['Duration: ',num2str(avgtime)]);
    text(180,topline+(separation*3),['Th: ',...
        num2str(thresh1),', ',num2str(thresh2)]);
    text(180,topline+(separation*4),['Segs/FFT: ',...
        num2str(segsPerFFT)]);
    text(180,topline+(separation*5),['Resolution: ',...
        num2str(FFTres)]);
    text(180,topline+(separation*6),['FFTs Avgd: ',...
        num2str(FFTAvgCount)]);

    pause;
  end; % end of if (nrf == ... refs vs primaries
  moreplots  = input(['More Plots (0=no, 1=yes) ? ']);
end;

==============================================================================

% psdecg.m rev 940811 fr
%
% matlab program to make power spectral density plots from ascii
% .qsd files (psd files from an .ecg file).
% There are 16 .ecg channels, we plot 8 at a time.
%
% get 6 char. Generic filename, Load info file
nam=input('Enter generic 6-char fcg filename in single quotes : ');
namei = [nam, 'I'];

eval(['load c:\army\armspcod\',namei,'.QSD']);
info = eval(namei);

% get the information from the info file
plotdate = [num2str(info(1)),'/',num2str(info(2)),'/',num2str(info(3))];
if (info(5) < 10);
  plottime = [num2str(info(4)),':0',num2str(info(5))];
else;
```

```
  plottime = [num2str(info(4)),':',num2str(info(5))];
end;
avgtime  = info(6);
FFTres   = info(7);
fpts = FFTres / 2 + 1;

% generate frequency vector
for i=1:fpts;
  freq(i)=(i-1)*208.333333/fpts;
end;

% Build .qsd file names, load files:
for ns=1:16;
  ns10 = int2str(fix(ns/10));
  ns1  = int2str(rem(ns,10));
  name(ns,:) = [nam, ns10, ns1]; % build file name
  eval(['load c:\army\armspcod\',name(ns,:),'.QSD']) % load file

% get the vectors for each trace
  plotXVector(ns,:) = freq;
  temp = abs(eval(name(ns,:)))/(10^(ns*3));
  plotYVector(ns,:) = temp';
end;

% draw the plot
semilogy(plotXVector',plotYVector');
titlee;
topline=1.0*10^5;
separation=-1;
text(180,topline*10^separation,['Date: ',plotdate]);
text(180,topline*10^(separation-2),['Time: ',plottime]);
text(180,topline*10^(separation-4),['Duration:',...
     num2str(avgtime)]);
text(180,topline*10^(separation-6),['Resolution: ',...
     num2str(FFTres)]);
pause;

=============================================================================

% psdmoco.m rev 940810 fr more coherence plots (refs wr to afet of ref)
%

% get 6 char. Generic filename, Load info file
nam=input('Enter generic 6-char fcg filename in single quotes : ');
namei = [nam, 'I'];

eval(['load c:\army\armspcod\',namei,'.PSD']);
info = eval(namei);

% get the information from the info file
plotdate = [num2str(info(1)),'/',num2str(info(2)),'/',num2str(info(3))];
if (info(5) < 10);
  plottime = [num2str(info(4)),':0',num2str(info(5))];
else;
  plottime = [num2str(info(4)),':',num2str(info(5))];
end;
nfets = info(6);
nrefs = info(7);
nthru = info(8);
avgtime  = info(9);
thresh1  = info(10);
thresh2  = info(11);
FFTres   = info(12);
fpts = FFTres / 2 + 1;
segsPerFFT = info(13);
```

```
FFTAvgCount = info(14);
tmp=14+nrefs+2;
% icohchan = info(tmp); % coherences of refs with which fet or ref?
icohchan = input(['Last calc. coherchan = ', int2str(info(tmp)),...
                  '; Select it or another: ']);
% generate frequency vector
for i=1:fpts;
  freq(i)=(i-1)*208.333333/fpts;
end;

% Coherence plots
% Build file names, load files:

if(icohchan <= nfets);
  namend1=[int2str(icohchan)];
  namend2=['.fsd'];
else;
  namend1=[int2str(icohchan-nfets)];
  namend2=['.rsd'];
end;
for nr=1:nrefs;
  namec(nr,:) = [nam,int2str(nr),namend1];        % build file name
  numc(nr) = info(14 + nr);       % get the plot ID
  eval(['load c:\army\armspcod\',namec(nr,:),namend2]) % load f

% get the vectors for each trace
  plotXVector(nr,:) = freq;
    temp = abs(eval(namec(nr,:))) - nr + 1;
  plotYVector(nr,:) = temp';
end;

% draw the plot
plot(plotXVector',plotYVector');

titlc;
topline=0.7;
separation=-0.25;
text(180,topline,['Date: ',plotdate]);
text(180,topline+(separation*1),['Time: ',plottime]);
text(180,topline+(separation*2),['Duration: ',num2str(avgtime)]);
text(180,topline+(separation*3),['Th: ',...
    num2str(thresh1),', ',num2str(thresh2)]);
text(180,topline+(separation*4),['Segs/FFT: ',...
    num2str(segsPerFFT)]);
text(180,topline+(separation*5),['Resolution: ',...
    num2str(FFTres)]);
text(180,topline+(separation*6),['FFTs Avgd: ',...
    num2str(FFTAvgCount)]);
pause;

============================================================================

% titl2.m rev 940812 fr
%
% matlab program to grid, title power spectral density plots from ascii
% .psd files , eig files, generated by fcghp572.for; also see psdf.m to
%  generate filenames, psd.m to make plots.)
%
if(nrf == 1)      % Case of References & Thru

      titleLine = [nam, ' PSD: References ',int2str(numr(1))];

  for nr=2:nrefs,
    titleLine = [titleLine,', ',int2str(numr(nr))];
  end;
```

```
%  titleLine = [titleLine,'; Thru: ',int2str(info(23))];
  titleLine = [titleLine,', Thru: ',int2str(nfets+nrefs+1)]; % fix armsp!
  title(titleLine);

elseif (nrf == 2) % Case of a Primary with Correcteds & Thru
  name=([nam,int2str(nf)])
  title([name,' PSD: Primary ',int2str(nf),', Corr1, Corr2, Source']);

elseif (nrf == 3)
  titleLine = [nam, ' Coherence w. Thru: Refs ',int2str(numc(1))];

  for nr=2:nrefs,
    titleLine = [titleLine,', ',int2str(numc(nr))];
  end;

  title(titleLine);
end

grid

=================================================================================

% titlc.m rev 940810 fr
%
% matlab program to grid, title power spectral density plots from ascii
% .psd files , eig files, generated by fcghp572.for; also see psdf.m to
%  generate filenames, psd.m to make plots.)
%
if(icohchan <= nfets);
  titleLine = [nam, ' Coherence, Refs w. Fet Ch. ',int2str(icohchan)];
elseif(icohchan <= nfets+nrefs);
  titleLine = [nam, ' Coherence, Refs w. Ref Ch. ',int2str(icohchan)];
else;
  titleLine = [nam, ' Coherence, Refs w. Thru Ch. ',int2str(icohchan)];
end;

title(titleLine);
grid;

=================================================================================

% titlee.m rev 940810 fr
%
% matlab program to grid, title power spectral density plots from ascii
% .psd files , eig files, generated by fcghp572.for; also see psdf.m to
%  generate filenames, psd.m to make plots.)
%
titleLine = [nam, ' Channels 1 - 16'];
title(titleLine);
grid
```